

Speedy Computer

costruisci programmi di
ADVENTURE
per il tuo computer



GRUPPO EDITORIALE JACKSON

PROGRAMMI PER
C 64 • SPECTRUM (48 K)
APPLE • VIC 20 • ZX81
BBC • TRS-80



costruisci programmi di
ADVENTURE
per il tuo computer

Jenny Tyler e Les Howarth

Progetto di Roger Priddy
Illustrazioni di Penny Simon,
Rob Mc Caig e Mark Longworth

La versione per ZX81 della "Casa Stregata" é di Chris Oxlade



Indice

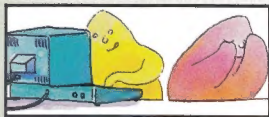
- 3 Presentazione di questo libro
- 4 Che cos'è un gioco d'avventura?
- 6 Dove iniziare
- 8 Disegnare un reticolo
- 10 Nascondere il tesoro
- 12 Oggetti utili
- 13 ... e come li si può utilizzare
- 14 Lo schema principale
- 16 Inserire i dati nel computer
- 19 La struttura del programma
- 20 Nell'interno del programma
- 28 Cambiare il programma
- 32 Il debugging della vostra avventura
- 33 Il listato della "Casa Stregata"
- 38 Modifiche per lo Spectrum (Timex 2000)
- 39 Versione per ZX81
- 46 Ulteriori suggerimenti ed indicazioni
- 47 Risposte agli enigmi
- 48 Indice analitico

NOTA SPECIALE

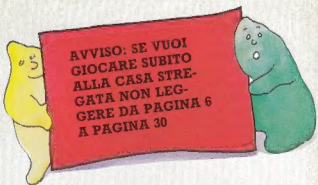
Per scrivere un gioco d'avventura dovete conoscere almeno un po' il linguaggio BASIC. Potrete comunque divertirvi con questo libro anche senza conoscere il BASIC, inserendo da tastiera e giocando con il gioco "LA CASA STREGATA", scritto apposta per questo testo. Se volete imparare o migliorare il vostro BASIC, vi sono altri due libri della JACKSON che vi potranno aiutare: "IMPARIAMO A PROGRAMMARE" "PRIMI PASSI IN BASIC"

Presentazione di questo libro

Scrivere un gioco d'avventura partendo da zero è un compito che spaventa, soprattutto se siete inesperti di programmazione dei calcolatori. Questo libro vi permette di iniziare quanto più gradatamente volete, fornendovi il listato di un gioco d'avventura che potrete inserire nel computer e con cui potrete giocare, effettuare variazioni e aggiunte a piacer vostro, oppure potrete usare come programma di base per crearvi la vostra avventura personale. Il gioco scritto per questo libro si chiama "La casa stregata", e potete trovarne il listato principale alle pag. 33-38. Questo listato girerà su qualsiasi computer che usi un BASIC di tipo Microsoft, compresi i modelli BBC, C64, e VIC 20 espanso, ed ha alcuni cambiamenti da apportare per poter girare su uno SPECTRUM 48K (o TIMEX 2000). Una versione speciale per ZX81 (o TIMEX 1000) è presentata con il listato alle pagine 39-45. Passate dirattamente a queste pagine se volete provare a giocare subito, prima di scoprire come funziona il programma. Questa può essere una buona idea se non avete mai giocato prima ad un gioco d'avventura.



Se siete abituati a leggere i listati dei programmi, probabilmente ricaverete qualche indizio sul funzionamento del programma mentre lo inserite da tastiera. Se voleste evitare che ciò accada, potreste convincere qualcuno ad inserirlo per voi. Essendo il programma piuttosto lungo e complicato, dovrete inserirlo con estrema attenzione. Vale la pena di battere lentamente alla tastiera, controllando ogni linea mentre la inserite, verificandone così la correttezza una volta per tutte. Potrete poi salvare il programma su nastro per la prossima volta che vorrete giocare o programmare una versione modificata del gioco.



**AVVISO: SE VUOI
GIOCARRE SUBITO
ALLA CASA STRE-
GATA NON LEG-
GERE DA PAGINA 6
A PAGINA 30**

Alle pagine 6-15 scoprirete come pianificare un gioco d'avventura, e alle pagine 16-27, come strutturarne e scriverne il programma. Potrete forse trovare piuttosto difficile questa parte del libro. Se è così non vi preoccupate, procedete passo dopo passo, lentamente e con attenzione fino ad essere sicuri di aver afferrato i concetti mano a mano che sono esposti, prima di proseguire oltre.

Mentre leggete queste pagine, può essere una buona idea quella di scrivere un programma vostro, per fare un po' di pratica, seguendo con attenzione ciascuna fase. Non vi preoccupate se non ne viene fuori un gioco particolarmente ben fatto; vi potrà comunque essere d'aiuto a capire come funziona il programma e vi potrà far vedere quali problemi occorre risolvere per poter scrivere un gioco d'avventura. Un buon gioco d'avventura deve essere pianificato con cura, per risultare interessante ed emozionante da giocare. Ricordate, non dovrete toccare il computer fino a quando non avete steso la trama del vostro gioco fino all'ultimo dettaglio. A pag. 46 potrete trovare altri suggerimenti ed indicazioni su come si scrive un'avventura, e a pag. 47 vi sono le risposte agli enigmi da risolvere che troverete nel libro. Dopo aver giocato qualche volta alla "Casa stregata", probabilmente vi verrà voglia di apportarvi dei cambiamenti. Alle pag. 28-31 troverete delle idee per realizzare la vostra versione del gioco.

Quali sono le regole?

Se avete già giocato a un gioco d'avventura, saprete già cosa aspettarvi dalla "Casa stregata". Se non lo avete mai fatto, vi serve solo sapere che il computer vi chiederà cosa volete fare, e voi glielo dovrete dire, con non più di due parole. È buona regola, durante il percorso del gioco, raccogliere qualsiasi cosa trovata che possa tornare utile per altri motivi, e di provare a usare queste cose in tutti i modi possibili per risolvere i problemi che incontrerete. Inserendo la parola "punteggio" da tastiera, scoprirete quanti punti avete fatto, e se avete vinto.

Che cos'è un gioco d'avventura?

Un gioco d'avventura è come un romanzo in cui il giocatore è il protagonista. Diversamente da un romanzo scritto in cui la sequenza degli avvenimenti è sempre la stessa, un gioco d'avventura è ogni volta diverso, poiché è il giocatore a scegliere cosa accade ad ogni momento. Fornendo al computer le istruzioni in risposta alle descrizioni che compaiono sullo schermo, il giocatore si avventura in un viaggio pericoloso in una terra sconosciuta. Lo scopo finale è quello di sopravvivere a qualunque pericolo si possa incontrare e tornare pieni di tesori.

Il primo gioco d'avventura venne scritto nel 1976 su di un grande elaboratore, all'Università di Stanford, negli USA, da William Crowther e Don Woods. È per lo più conosciuto con i nomi di Colossal Cave, Colossal o più semplicemente Adventure, ed oggi ne esistono versioni disponibili su molti home-computers. Quel gioco venne scritto nel linguaggio scientifico dei computer, il Fortran, che a differenza del Basic, non riconosce le parole di testo. Tutti i dati per quel gioco dovettero essere indicizzati ed immagazzinati su memoria a disco. Le prime persone che giocarono ai giochi d'avventura furono dei professionisti del settore dei calcolatori, poiché allora i computer da casa non esistevano. Oggi si usa ancora includere una versione del gioco di Crowther e Woods in molti grandi sistemi ad elaboratore, per dimostrare alla gente non abituata al computer, che questi ultimi possono anche dimostrarsi "di compagnia". Queste avventure basate su memoria a dischi spesso occupano più di 250 K di memoria, e sono molto complicate da giocare.

Le avventure per i personal computer

Sono state scritte molte avventure dopo quella originaria. Forse le più famose sono quelle scritte da Scott Adams, un programmatore americano, che è stato il primo a realizzare una versione di Adventure per microcomputer. Questa si chiamava Adventure land (terra d'avventure), e girava sul TRS-80. Altre avventure di Scott Adams da tenere presente sono: Pirate Adventure (Avventura dei pirati), The count (Il Conte), e Pyramid of doom (La piramide della dannazione).

Il termine "avventura" oggi viene usato per descrivere una vasta gamma di giochi diversi. Il gioco in questo libro rappresenta un'avventura tradizionale, composta solo di

testi, e basata sul tipo di gioco realizzato da Crowther e Woods. Il giocatore assume il ruolo del protagonista, ma non è dotato di caratteristiche particolari, come nei giochi dal vivo, in cui si interpretano delle parti. Il giocatore usa la propria intelligenza, l'astuzia, e così via, e non quella di un personaggio che gli venga assegnato da interpretare all'inizio, come succede in tali giochi. Come negli scacchi, le avventure tradizionali sono giochi che tengono impegnata la mente, e comportano la risoluzione di enigmi, piuttosto che non la prontezza di riflessi o la fortuna.

Avventure grafiche

Il gioco d'avventura originale non faceva uso di disegni, ma si affidava piuttosto all'immaginazione del giocatore per dare vita ai mostri e agli altri terrori che fanno parte del gioco. Alcuni pensano che un'avventura che presnti anche dei disegni non sia una vera avventura, per quanto oggi vi siano dei giochi grafici di avventura per microcomputer molto ben fatti. Se disponete di sufficiente memoria nel vostro computer potrete aggiungere delle routine grafiche alle vostre avventure, o al programma "La casa stregata" riportato in questo libro. Qui non vengono riportate spiegazioni su come fare ciò poiché le istruzioni grafiche variano molto da computer a computer.





Che tipo di programma è?

Un programma d'avventura è, in realtà, una particolare forma di archivio dati (database). Un database è un sistema di archiviazione per computer, che immagazzina informazioni e permette che queste vengano richiamate in diversi modi, e può avere molti usi. Un programma d'avventura è un database "interattivo".

Il giocatore ci si muove dentro, cambiando o "aggiornando" le informazioni. Andando avanti nella lettura, vedrete come certe parole vengano usate come "chiavi" per avere accesso a certe parti di informazione. Questa tecnica può essere usata, per database "seri", per limitare l'accesso a certe informazioni. Potrete imparare alcune utili tecniche di programmazione scrivendo un gioco d'avventura. Essendo questo un tipo di programma abbastanza complicato, mostra quanto sia importante pianificarlo prima in dettaglio. Se riuscite a trovare un modo in cui il computer possa trattare qualsiasi tipo di input, per quanto insensato possa essere, allora sarete in grado di scrivere programmi che non si bloccano o autodistruggono. Molte delle avventure che potete comprare su cassetta sono scritte, almeno in parte, in linguaggio macchina. Ciò permette di inserire ancora più informazioni nel computer, e fa girare il gioco più velocemente. Se conoscete un po' il linguaggio macchina, potreste sperimentare delle aggiunte di routines in linguaggio macchina ai vostri giochi d'avventura.



Ora voltate pagina ed iniziate a pianificare il vostro gioco d'avventura.

Dove iniziare

Scrivendo un gioco d'avventura, inventate un mondo fantastico dove siete voi a stabilire tutte le regole. Siete voi a decidere lo scenario, che tipi di creature e di cose, e quello che tali creature e cose possono fare. Il vostro mondo può essere una città extraterrestre, per esempio, o un palazzo sotterraneo abitato

da elfi, maghi e gnomi, oppure un castello misterioso che sia la residenza di draghi e altri strani mostri. Ancora, potrebbe essere ambientato in tempi passati, coinvolgendo personaggi e fatti storici reali. Molte avventure prevedono l'uso di magie di diverso genere. Potete decidere quanto far aderire il vostro mondo alle regole del mondo reale, e quanta magia vi sia permessa. Qualunque cosa scegliate di fare, assicuratevi che le regole del vostro gioco siano logiche, o i giocatori potrebbero trovare il gioco privo di senso. Avendo stabilito uno scenario per il vostro mondo d'avventura, dovrete decidere quale sia lo scopo del vostro gioco. Il giocatore potrà dover fuggire da un certo posto, o ritornare in un certo posto, carico di tesori, oppure dovrà salvare qualcuno, o ancora, scoprire qualche luogo sconosciuto per portarvi a termine una missione.

Stabilire le locazioni

Le aree o stanze attraverso le quali un giocatore "si muove" nel corso di un'avventura si chiamano locazioni. Più avanti nel libro vedrete come queste vengano numerate per poterle inserire nel computer. Per adesso, vi servirà ricordare che il numero di locazioni che potete avere dipende dalla quantità di memoria a disposizione del vostro computer. Più locazioni rendono il gioco più interessante, ma vi lasciano con meno spazio in memoria per tutte le loro descrizioni. Il gioco presentato in questo libro ha 64 locazioni dotate di brevi descrizioni.


Una locazione può trovarsi in ambiente chiuso o luogo all'aperto, e può essere una stanza, una caverna, a metà di un corridoio, un'area in una foresta, in mezzo ad un campo o dovunque altro vi piaccia. La cosa migliore è decidere il numero di locazioni subito, e mantenerlo al valore deciso, in quanto esso condiziona tutta la struttura del gioco.

Tracciare una mappa

Il passo successivo è quello di disegnare un abbozzo di mappa del vostro mondo di fantasia. Potrà anche non essere accurata fino al minimo dettaglio, ma dovrà mostrare l'intero ambiente nelle giuste proporzioni. Così facendo, pensate anche a quali possano essere dei buoni nascondigli per tesori e altri oggetti che potranno servire al protagonista dell'avventura.

Ecco riportato un abbozzo di mappa per l'avventura della "Casa stregata", scritta per questo libro.





"La casa stregata" è un gioco ambientato in una strana casa sul margine di una ripida scogliera. Le sue torrette si profilano contro un cielo sempre tetro. Nessun muro sembra essere diritto, nessun angolo sembra essere di novanta gradi. Non è una sorpresa che non vi abiti nessuno — beh, nessun essere umano, per lo meno ...

La gente dice che l'uomo più ricco del mondo vi abbia passato i suoi ultimi giorni. Stranamente, nessuno è mai riuscito a trovare il suo corpo ...

Idee per soggetti

Se non vi vengono idee per scrivere il vostro gioco d'avventura, pensate a films o a programmi televisivi che avete visto, o a libri che avete letto. Ricordate, perciò, che se state scrivendo un programma di gioco con l'intenzione di farne commercio, di non seguire troppo fedelmente la trama del film o del libro, o usare gli stessi nomi dei personaggi, per via dei diritti d'autore. Eccovi alcune idee per soggetti d'avventura.

UNA STORIA POLIZIESCA — Il giocatore fa la parte di un detective che indaga su un terribile crimine. Scopo di questo gioco è quello di tornare alla centrale di Polizia con tutte le prove per scoprire il colpevole (gli oggetti che costituiscono le prove sono i "tesori" da trovare).

UN'AVVENTURA PREISTORICA — Il giocatore è tornato indietro nel tempo fino agli uomini delle caverne. Il suo scopo è quello di tornare al giorno d'oggi portando con sé La Pietra, oggetto di immenso potere. Il protagonista deve costruirsi da solo le armi e qualsiasi cosa gli possa servire. Tra gli ostacoli che incontrerà vi sono animali preistorici e le magie delle caverne.

TERRORE NEL TEMPIO — Le antiche rovine di un tempio costruito da un popolo misterioso, e da lungo tempo estinto, contengono il segreto della vita eterna. Il solo ascoltare le cose che si dice vi siano successe fa rizzare i capelli. Scopo del gioco è riuscire a scappare dal tempio con il segreto.



Disegnare un reticolo

Il primo passo per far diventare un gioco per computer il vostro mondo d'avventura, è quello di trasferire la vostra mappa su di un reticolo. Avrete bisogno di un riquadro del reticolo per ogni locazione, perciò per le sue 64 locazioni il gioco della "Casa Stregata" ha bisogno di un reticolo 8x8.

Questo reticolo diventerà poi il riferimento principale per la vostra avventura, per cui disegnate il più ampio e chiaro possibile. Alla fine esso mostrerà tutte le locazioni ed i percorsi in ingresso e in uscita da ciascuna, e tutti i tesori e gli oggetti usati nel gioco.

Numerate ciascuna locazione, iniziando dal riquadro dell'angolo superiore sinistro. La maggior parte dei computer inizia i conteggi da 0, per cui scrivete 0 come numero della prima locazione.

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Potreste voler cambiare la posizione che un giocatore può seguire, per cui iniziate a disegnare la vostra mappa con tratto leggero a matita, sul reticolo. Definite ciascuna locazione con una breve descrizione, ad esempio "cantina buia" o "stanza polverosa", poi decidete quali possono essere gli ingressi e le uscite da ciascuna locazione. Il modo più comune per indicare tali vie è quello di usare i punti cardinali — il nord verso la parte alta del reticolo, il sud verso il basso, l'est verso destra e l'ovest verso sinistra.

Includendo scalinate, scale a pioli o botole nella vostra descrizione, potrete anche usare "in alto" e "in basso" come vie da seguire rispettivamente verso l'alto o verso il basso, al posto dei punti cardinali. Questo rende il gioco più interessante, senza dover ricorrere ad un vero reticolo tridimensionale.

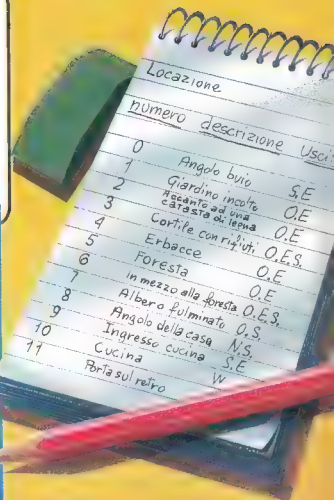
Stabilire i percorsi

Contrassegnate le uscite da ciascuna locazione sul vostro reticolo, come in figura. Notate che alcune locazioni su questo reticolo presentano dei percorsi a senso unico, per cui il giocatore non può tornare indietro. Assicuratevi che vi siano dei validi motivi per usare tali percorsi nella vostra avventura. Nella "Casa Stregata" la porta d'ingresso si chiude di colpo a chiave, alle spalle del giocatore, non appena questi è entrato, così che non possa uscire di nuovo. La palude alle locazioni 53 e 54 costituisce anch'essa un percorso a senso unico poiché la



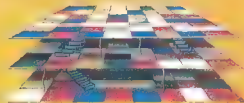
barca vi si impantana dentro. Quanti possibili percorsi a senso unico vi vengono in mente, guardando la figura? Quando avete deciso a proposito dei percorsi, ritracciate a penna i muri ■ le scalinate che completano il quadro, ■ compilate una tabella come quella mostrata più avanti, facendo un elenco con il numero della locazione, la descrizione di questa (anche se non è quella definitiva), ■ le sue uscite. Questa tabella aiuterà molto quando inizierete a scrivere il programma.

4 MACCHIA DI ERBACCE E	5 FORESTA E	6 IN MEZZO ALLA FORESTA S E	7 ALBERO FULMINATO S
12 STANZA POLVEROSA S E	13 STANZA DELLA TORRE SUL RETRO O	14 RADURA N	15 SENTIERO S
20 SCALA A CROCCIOLO N Su O Giù	21 AMPIO CORRIDOIO S	22 SCALINI SCIVOLOSI Su S Giù	23 CIMA DELLA SCOGLIERA N S
28 SALONE E	29 STANZA DEI TROFEI N S	30 CANTINA N S	31 SENTIERO SULLA SCOGLIERA N S
37 RIPIDE SCALE MARMO N Su	38 SALA DA PRANZO N	39 IN FONDO ALLA CANTINA N	39 SENTIERO SULLA SCOGLIERA S
44 STANZA PIENA DI RAGNATELE N S E	45 CAMERA FREDDA O	46 STANZA SPETTRALE E	47 SENTIERO SULLA SCOGLIERA ACCANTO A UNA PALUDE N S
52 PASSAGGIO SUPERIORE O	53 PALUDE ACCANTO ALLE MURA S	54 PALUDE O S	55 SENTIERO MELMOSO N O
60 DETRITI O	61 CALCINACCI N	62 ARCO DI PIETRA N	63 TERRENO FRANGENTE IN CIMA ALLA SCOGLIERA O



Per aiutarvi a capire come questo tipo d'informazione vada inserito nel programma, potrete completare questa tabella usando la mappa a pag. 14-15 e confrontarla con il listato a pagina 37.

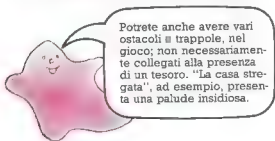
Giochi in 3 dimensioni



È possibile costruire vere e proprie avventure in 3 dimensioni, poste su diversi livelli, come i piani di una casa. Per far questo, avrete bisogno di due o più reticoli interconnessi. Tali giochi necessitano di molta memoria (probabilmente almeno 48K perché ne valga la pena), e i relativi programmi possono diventare molto complicati da scrivere.

Nascondere il tesoro

Avendo tracciato la mappa del vostro scenario d'avventura dovete ricominciare a pensare a quel che il giocatore dovrà fare in tale ambiente. In molte avventure il giocatore deve trovare oggetti di valore di vario genere, e portarli in qualche posto particolare. Questi potrebbero essere dei "veri" tesori, quali oro e gioielli, o potrebbero anche consistere in piani segreti o documenti o prove che servano a risolvere un crimine. Ancora, se lo scopo del vostro gioco è quello di salvare qualcuno, considerate la persona da salvare come un "tesoro" nascosto.



Potrete anche avere vari ostacoli e trappole, nel gioco; non necessariamente collegati alla presenza di un tesoro. "La casa stregata", ad esempio, presenta una palude insidiosa.

Aggiunta di "arredi"

Dovrete decidere, quindi, che tesori avere e dove nascondarli. Nascondere i tesori probabilmente comporterà il dover includere degli "arredi" al vostro ambiente. Questi possono essere dei mobili, dei tappeti, capi di vestiario o altro, che il giocatore possa aprire ed esaminare, ma che non possano essere rimossi dalla locazione in cui si trovano. Nella "Casa Stregata" uno degli "arredi" è costituito da una bara.

Alcune idee a cui pensare

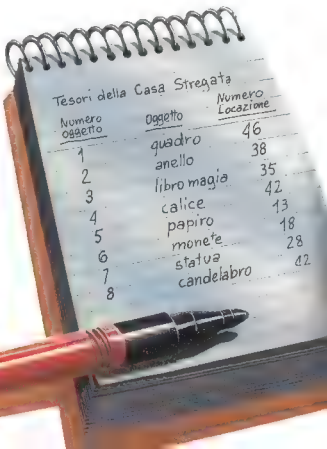
Riuscite a pensare ad alcuni tesori che possano andar bene per i seguenti ambienti di gioco?

1. Il quartier generale di un'associazione criminosa internazionale.
2. Un pianeta lontano che è noto essere tecnologicamente più evoluto della terra.
3. Un'installazione segreta per la ricerca scientifica.

Fatto questo, riuscite ad immaginare dei validi ostacoli al raggiungimento di tali "tesori"?

Creare delle difficoltà al giocatore

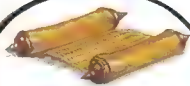
A questo punto dovete pensare a quali difficoltà il giocatore dovrà risolvere per poter trovare i tesori e portarseli via. Quanto più i problemi che porrete al giocatore saranno ben congegnati ed originali, tanto più interessante sarà lo svolgimento del gioco. Le soluzioni a molti dei problemi dovranno essere ottenute con altri oggetti che il giocatore avrà dovuto trovare ed utilizzare nel modo giusto. Nel paragrafo che segue troverete altre indicazioni sugli oggetti "utili". Anche qui, fate una tabella dei vostri oggetti di valore e numerateli, cominciando stavolta da 1. (Capirete il perché leggendo a pagina 28). Elencate gli oggetti in ordine decrescente di valore, poiché questo vi servirà più avanti per stabilire il sistema di punteggio. Questa lista sarà il primo elenco di parole che il vostro computer dovrà poter riconoscere. Segnatevi anche quali sono gli ostacoli che si incontrano prima di arrivare a ciascun tesoro. Potrete avere ad esempio un mostruoso guardiano, od un gnomo pazzo armato di ascia. I tesori potranno trovarsi in cassette chiuse a chiave, e in casserforti. Potranno essere impossibili da trasportare senza un contenitore di qualche genere, e questo sarà nascosto altrove. Qui di seguito vi presentiamo alcuni enigmi a cui pensare, come esempio.



Tesori della Casa Stregata		
Numero oggetto	Oggetto	Numero Locazione
1	quadro	46
2	anello	38
3	libro magia	35
4	calice	42
5	papiro	13
6	monete	18
7	statua	28
8	candelabro	42



6. Sacco pieno di monete d'oro: avete bisogno di fare luce.



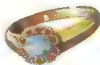
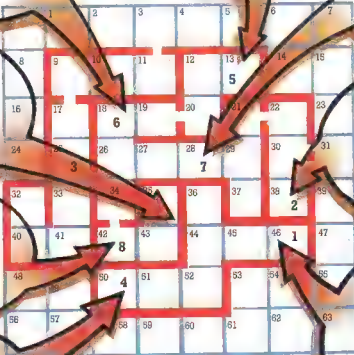
5. Antico Papiro: custodito da pipistrelli.



7. Statua d'ebano: avete bisogno di luce.



3. Libro di magia nascosto nella stanza segreta dietro un falso muro.



2. Anello di diamanti nascosto nella bara.



8. Candelabro d'oro



1. Quadro a olio: protetto da una barriera magica e da una porta chiusa.

4. Calice: custodito da spiriti paralizzanti e da una porta chiusa.

Segnate i vostri tesori sullo schermo, usando i numeri che avete assegnato ■ ciascuno sulla vostra lista. Poi scrivete sulla lista, accanto al nome dell'oggetto, anche quello della locazione (potrete avere solo 1 oggetto in ciascuna locazione).

Enigma poliziesco

In questo gioco gli oggetti di valore sono:

1. Un capello rosso
2. Un filo di lana marrone
3. Un'impronta sul terreno
4. Una serie di impronte digitali
5. Una macchia di sangue
6. Un pesante bastone di legno

Le difficoltà che si incontrano per arrivare a tali oggetti sono:

1. Il capello potrà andare perduto, o danneggiarsi se lo prendete così com'è. Esso si trova sulla manica di un cappotto.
2. Il filo invisibile ad occhio nudo, si trova sul bordo interno di un cassetto

chiuso a chiave.

3. L'impronta si trova su di un'aiuola, sotto ad una finestra.
4. Le impronte digitali sono sulla superficie di un grande tavolo accanto al cadavere. Esse sono invisibili.
5. La macchia di sangue è sul tappeto.
6. Toccando il bastone si cancelleranno le impronte digitali che eventualmente vi si trovino.

Che soluzioni riuscite a trovare? Vi sono dei suggerimenti a pag. 47, benché le vostre soluzioni potrebbero essere migliori.

Oggetti utili ...

Per aiutare il giocatore a risolvere i problemi che gli ponete, dovrete includere degli strumenti, armi, ed altri oggetti utili nel vostro gioco. Il giocatore deve trovare gli oggetti appropriati ed usarli nel modo giusto per aggirare gli ostacoli. Potrete mettere alla prova l'ingegnosità del giocatore, non includendo oggetti il cui uso fosse troppo ovvio; al posto di una chiave, ad esempio, potrete includere una forcina per capelli o una graffetta, per aprire una porta chiusa. Potrete anche complicare le cose, nascondendo ad esempio una torcia elettrica in una locazione, e le pile per farla funzionare in un'altra. Il giocatore dovrà così trovare entrambe le cose prima di poterle usare.

Aggiungete questi oggetti alla lista di parole iniziata con i tesori. Non dimenticate, peraltro, che anche alcuni tesori possono essere utilizzati in qualche modo (questo non significa che li dobbiate elencare due volte).

Problemi e soluzioni della "Casa Stregata"

Ecco le soluzioni ai problemi del gioco "La Casa Stregata", e gli oggetti richiesti per ottenerle:

Problemi	Soluzione	Oggetti necessari
Troppo buio per poter vedere	Mettere la candela nel candeliere; accenderla con un fiammifero	Candela (nascosta nel cassetto della scrivania) Fiammiferi Candeliere (già nella lista dei tesori)
Pipistrelli	Usare lo spray mortale	Flacone di aerosol
Stanza stregata	Abbatte il falso muro	Ascia
Porta sprangata	Aprire la serratura	Chiave
Fantasma paralizzanti	Risucchiarli nell'aspirapolvere	Aspirapolvere portatile, pile
Barriera magica	Usare incantesimo magico	Libro delle magie (già nella lista dei tesori)
Palude	Attraversarla in barca (può essere usata una volta sola perché poi si impantana nel fango)	Barca
Finestra sbarrata	Scavarci intorno per togliere le sbarre	Vanga
Bara di legno	Aprire il coperchio	Nessuno

Decidete dove collocare gli oggetti, e inseriteli nello schema principale. Li renderete meno evidenti se li metterete in posti dove ci si aspetterebbe di trovarli, ad esempio un coltello in cucina, un libro nello studio o nella biblioteca, un'ascia vicina alla catasta di legname. Potrete voler aggiungere altri arredi (vedere paragrafo precedente), a questo punto della pianificazione. "La Casa Stregata" presenta una scrivania in cui è nascosta la candela. Ricordatevi di non mettere oggetti in posti impossibili. È inutile mettere la torcia dietro una porta sprangata e la chiave in una stanza buia. Mettete il numero della locazione accanto a ciascun oggetto. Aggiungete alla lista qualsiasi altra parola (purché non sia un verbo) che vorrete far riconoscere al computer, ad esempio nord, sud, fantasmi, pipistrelli, bara (ricordatevi di includere tutti gli arredi).

Elenco Oggetti della casa stregata

Numero Oggetto	Oggetto	Numero locazione
9	Fiammiferi	10
10	Aspirapolvere	25
11	Pile	26
12	Vanga	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26
13	Ascia	10
14	Barca	25
15	Barca	26
16	Aerosol	10
17	Candela	25
18	Chiave	26
19	Nord	10
20	Sud	25
21	Ovest	26
22	EST	10
23	In alto	25
24	In basso	26
25	Porta	10
26	Pipistrelli	13
	Fantasma	52
	Cassetto	43
	Scrivania	43
	Cappotto	32

... e come li si può utilizzare

Ora che avete deciso quali oggetti inserire nel vostro gioco, che cosa farete fare con essi al giocatore? Qui dovrete fare una lista di verbi e delle corrispondenze tra questi e gli oggetti a cui si riferiscono. Tale lista deve prevedere anche verbi "di moto" cosicché i giocatori possano dare istruzioni al computer riguardo a dove vogliono spostarsi.

Molti programmi di avventura sono costruiti in modo che il computer accetti dal giocatore dei comandi composti da non più di due parole. Il computer confronta la prima parola con una lista di verbi che avete messo in memoria, e la seconda con gli oggetti e le direzioni a cui avete associato il verbo. Molto dell'interesse nello scrivere delle avventure sta nel provare a considerare tutte le combinazioni

di verbi ed oggetti che il giocatore possa voler tentare, e nel decidere la conseguente azione o risposta che il computer dovrà fornire. Anche chi fa programmi di tipo commerciale si trova a dover ragionare allo stesso modo, per evitare che il programma si sblocchi a causa di una risposta non prevista dell'utente. Per trattare con verbi (e oggetti) che il computer non trova in memoria, potrete includere delle risposte vaghe, del tipo: "Far che cosa con (l'oggetto)?". Raggruppate insieme i verbi che hanno significato analogo, ad esempio prendere e pigliare. Potrete così risparmiare spazio in memoria, mandando il programma alla stessa subroutine in entrambi i casi.

Lista dei verbi della "Casa Stregata"

Numerate i verbi a partire da 1. (Il computer usa il valore zero per indicare "verbo non trovato", (vedi pag. 16). Le parole "AIUTO" ed "INVENTARIO" (o "POSSESSO?"), sono caratteristiche dei giochi d'avventura, includete anche queste nella lista di verbi. È utile aggiungere forme abbreviate di "VA A NORD", etc., (vedi sotto verbi 4-9). Queste riducono lo sforzo di battitura del giocatore, rendendo più veloce il gioco.

N°	Verbo	Riferimento	Azione (eventuale condizione da soddisfare, tra parentesi)
0	—	—	Usato per indicare "verbo non trovato".
1	AIUTO	—	Elenca tutti i verbi noti al computer.
2	POSSESSO?	—	Elenca tutti gli oggetti che il giocatore ha con sé.
3	ANDARE	DIREZIONI	Muove la posizione.
4	N	—	Forma breve per "ANDARE A NORD".
5	S	—	Forma breve per "ANDARE A SUD".
6	■	—	Forma breve per "ANDARE A OVEST".
7	E	—	Forma breve per "ANDARE A EST".
8	A	—	Forma breve per "ANDARE IN ALTO".
9	B	—	Forma breve per "ANDARE IN BASSO".
10	PIGLIARE	OGGETTI	Raccogliere oggetti e portarli con sé. (L'oggetto deve essere nella locazione).
11	PRENDERE	OGGETTI	Lo stesso di "PIGLIARE".
12	ESAMINARE	QUALSIASI COSA	Rivelare oggetti nascosti.
13	APRIRE	PORTA: SCRIVANIA	Aprire una porta o un cassetto (il giocatore deve avere una chiave per la porta).
14	LEGGERE	LIBRI INCANTESIMO	Rivelare indizi scritti (il giocatore deve avere il libro degli incantesimi, o trovarsi nella biblioteca).
15	DIRE	QUALSIASI PAROLA	Pronuncia parole scritte "ad alta voce", es.: lancia un incantesimo.
16	SCAVARE	—	Scavare una buca (deve avere una vanga e trovarsi nel cortile).
17	BRANDIRE	ASCIA	Abbatte il falso muro (deve avere l'ascia e trovarsi nello studio).
18	SALIRE	CORDA	Salire o scendere lungo la corda.
19	SCENDERE	—	—
20	ACCENDERE	CANDELA	Accendere la luce (deve avere la candela, i fiammiferi ed il candeliere).
21	SPEGNERE	CANDELA	Spegne la luce (deve avere la candela accesa).
22	SPRUZZARE	AEROSOL	Scaffiare i pipistrelli dalla torre sul retro (deve avere l'aerosol).
23	USARE	ASPIRAPOLVERE	Risucchia i fantasmi (deve avere l'aspirapolvere e le batterie).
24	DISSERRARE	CHIAVE; PORTA	Aprire la porta (deve avere la chiave).
25	LASCIARE	QUALSIASI NOME	Lasciare giù un oggetto (deve avere l'oggetto).
26	PUNTEGGIO	—	Stampa il punteggio.

Lo schema principale

Il vostro schema principale e le tabelle che avete preparato contengono tutte le informazioni, e i dati, di cui il vostro programma ha bisogno. Ecco lo schema principale completo per "La Casa Stregata" (Non vi preoccupate se il vostro schema principale non appare così elaborato). Nel prossimo paragrafo troverete come inserire questi dati nel vostro computer. Prima di toccare il computer, però, assicuratevi di aver pianificato il vostro gioco fino all'ultimo dettaglio.

Problemi interessanti in un'avventura

Ecco alcune delle situazioni in cui i giocatori potrebbero trovarsi nel corso di un'avventura. Quante possibili soluzioni riuscite a trovare per ciascuna? A pagina 72 potrete trovare alcuni suggerimenti.

1. Siete intrappolati in una stanza di tre metri per tre. Non ci sono porte. Vi è uno spesso tappeto.
2. Entrando in una stanza, vi prende un senso di estrema sonnolenza. Avete con voi un piccolo ma pesante zaino, e un fazzoletto.
3. Siete sui bastioni di un castello. Sotto di voi un'orda di schiavi infuriati, e dietro di voi dei soldati armati. Avete un rotolo di papiro in mano.
4. Siete stati invitati a pranzo dal malvagio super-bandito. Egli vi ha portato via tutte le armi. Al momento del dolce, egli vi mostra il telecomando della sua arma per decimare il mondo.



Riuscite a pensare ad altre situazioni interessanti per delle avventure (e alle relative soluzioni, naturalmente)?

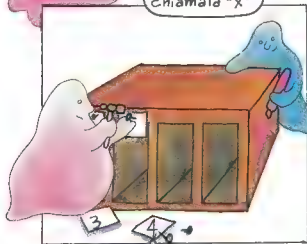




Inserire i dati nel computer

Ora avete tutti i dati per la vostra avventura scritti su vari fogli di carta. Il problema successivo è quello di stabilire come inserirli nella memoria del computer. Il computer deve avere i dati immagazzinati in modo tale da poter accedere a ciascuno rapidamente, e da poterli aggiornare facilmente durante l'evolversi del gioco. Per permettere ciò dovrete predisporre delle aree di memoria del computer per l'immagazzinamento dei dati, chiamate "matrici". Una matrice unidimensionale può visualizzarsi pensando ad una fila ordinata di caselle della posta, o di cassette di un archivio. Voi attribuite a ciascuna matrice un nome, e a ciascuna sua "casella" un numero, cosí che il computer possa accedere alla "casella" della memoria che gli

Questa è una matrice chiamata "X"



richiedete, quando vi riferite ad essa nel vostro programma. Prima di poter dare al computer dei dati, dovrete decidere quanto dovrà essere grande ciascuna matrice, e dire al computer di predisporre e di attribuire un nome allo spazio di memoria necessario. Questo si chiama "dimensionamento" della matrice, e in linguaggio "BASIC" si scrive DIM.

Le matrici per 'La Casa Stregata'

"La Casa Stregata" necessita delle seguenti matrici per contenere i dati. Vi serviranno comunque matrici simili, qualunque sia il tema della nostra avventura.

1. Una matrice di dati per contenere le descrizioni delle locazioni. Questa deve avere 64 caselle (una per ogni locazione). Noi l'abbiamo chiamata $D\$()$, e abbiamo numerato le caselle da 0 a 63 come sullo schema principale.

2. Una matrice per contenere l'informazione sui percorsi che il giocatore può seguire tra una locazione e l'altra. Questa viene chiamata

Il segno \$ indica che si tratta di una matrice di stringhe di caratteri, ovvero fatta per immagazzinare lettere e parole.

Questa matrice viene predisposta nel programma scrivendo DIM $D\$(63)$.

$R\$()$. Deve essere della stessa dimensione ■ numerata allo stesso modo di $D\$()$.

3. Una matrice per gli oggetti ■ e le altre parole sulla tabella da voi compilata. Dimensionando questa matrice con DIM $O\$(W)$ dove W è il numero di parole della lista, il computer predisporrà una matrice con una casella per ciascuna parola ed una casella vuota. Questo perché la numerazione inizia normalmente da zero e termina con il numero dichiarato nella istruzione DIM. Ciò torna utile perché la casella vuota può essere utilizzata per "parola non trovata". Ad esempio se $W = 4$, la matrice apparirebbe come questa: DIM $O\$(4)$ predisporrà 5 caselle numerate da 0 a 4.

4. Una matrice di verbi. Questa necessita di una casella per ciascun verbo ed una casella in più per "verbo non trovato". Viene chiamata $V\$()$, ■ deve essere dimensionata con DIM $V\$(V)$, dove V è il numero di verbi compresi nella vostra lista.

Ecco il motivo per cui le liste di verbi e di parole sono state numerate partendo da 1 e non da 0, come le locazioni.

QUANTO VALE W

ORA VALE 4

VARIABILI

Usando nomi di variabili quali V e W, rende piú facile cambiare il numero di parole ■ verbi nel seguito.

Altre matrici

Le locazioni, i percorsi, i nomi ed i verbi non sono le uniche informazioni che vanno immagazzinate nel computer. Servono anche matrici dove inserire informazioni del tipo: dove si trovano gli oggetti, quali oggetti sono in possesso del giocatore e, ad esempio, se la luce è accesa o meno.

Per far questo, non c'è bisogno di immagazzinare nuovamente le locazioni o gli oggetti. Queste informazioni in più possono essere inserite come numeri per risparmiare spazio in memoria, ad esempio: l'oggetto 9 è nella locazione 10.

La matrice $L()$ mostra in quale locazione si trova ciascun oggetto. Servono solo caselle per gli oggetti "trasportabili", quali ad esempio la chiave, non per gli arredi o altro. Se il numero degli oggetti trasportabili è G , allora questa matrice si dimensiona con $DIM L(G)$.

La matrice $C()$ serve per le informazioni su quali oggetti sono in possesso del giocatore. Anche qui servono solo caselle per gli oggetti trasportabili, per cui il dimensionamento avviene con $DIM C(G)$.

Indicatori

Le matrici numeriche non richiedono il segno \$ dopo il nome.

$L()$

Oltre a tener conto degli oggetti che il giocatore possiede, il computer deve poter registrare altri cambiamenti che avvengono durante il gioco, ad esempio, se la candela viene accesa, la porta chiusa a chiave, o la chiave resa visibile.

Questo si fa usando una matrice $F()$ di indicatori, o "flags", che contiene W caselle, ovvero una per ogni oggetto. Scrivendo 1 o 0 in queste caselle, il computer può tener conto dello stato in cui un oggetto si trova. Lo 0 viene usato per lo stato normale o "inattivo"

dell'oggetto, quale: luce spenta, o oggetto visibile. L'1 indica lo stato "attivo", o non normale, quale: luce accesa, o oggetto invisibile.



Sapevate che i computers hanno dei registri di flags nelle loro CPU, che funzionano proprio come quelli di questa matrice? I computers li utilizzano per tenervi l'informazione su quello che avviene nel corso dell'esecuzione di un programma.

Perché non usare matrici a due dimensioni?

Se avete avuto a che fare con le matrici prima d'ora, potrete esservi chiesti perché vengano usate matrici unidimensionali per le descrizioni o i percorsi della "Casa Stregata", invece di matrici a due dimensioni (che possono visualizzarsi come file

$A()$

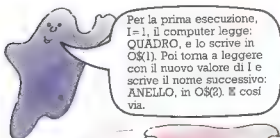


sovrapposte di caselle allineate). Il motivo è che le matrici ad una dimensione occupano un po' meno spazio in memoria. Potreste usare matrici a 2 dimensioni, se volete, nel qual caso le dimensionerete con $DIM D(8,8)$, e $DIM R(8,8)$.

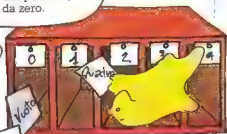
Mettere i dati nelle matrici

Avendo predisposto le aree di immagazzinamento con i loro nomi nella memoria del computer, dovete ora dirgli che cosa scriverci. Un modo per far questo è quello di elencare i dati in ordine, ed ordinare al computer di leggerli ad uno ad uno, scrivendoli uno alla volta nelle caselle di una matrice. (*). Ecco come si fa, con istruzioni BASIC:

```
DIM O$(W)
DATA IL QUADRO, L'ANELLO, INCANTESIMI
MAGICI, IL CALICE, ETC.
FOR I=1 TO W
READ O$(I)
NEXT I
```



Il ciclo inizia contando da per le matrici degli oggetti, dei verbi e per le matrici numeriche, quindi il computer lascerà vuota la casella zero. Per le matrici delle locazioni e dei percorsi, il ciclo inizia da zero.



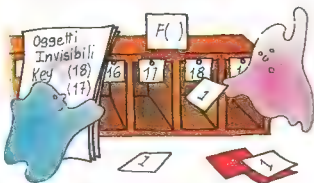
I singoli dati si scrivono separati da virgole.



Questo è il ciclo di READ (lettura) dei termini DATA in O\$(). Guardate il listato del programma alle pagine 36 e 37, alle linee 1600-2100, e cercate di individuare i cicli di lettura dati per le altre matrici.

Dati per gli indicatori

I dati per la matrice di indicatori F() consistono solo di uni e zeri. Gli oggetti che all'inizio del gioco sono invisibili prestano l nella casella di F() ad essi relativa. Quando il giocatore li scopre, l'indicatore cambia in 0. Tutti gli altri oggetti all'inizio sono contrassegnati con zeri. Voi dovrete quindi solo dire al computer quali caselle F() devono



contenere l. Lasciare le altre vuote è lo stesso che scrivervi zero. Il modo più semplice per riempire questa matrice è mostrato alla linea 2090 a pag. 37. Potreste aver notato che certe caselle di F() non vengono usate perché alcuni oggetti non cambiano di stato. Questi indicatori in più possono essere usati per altre cose. Per esempio, nella "Casa Stregata", la casella F(14) (indicatore della corda) è usata per indicare se il giocatore è salito sull'albero. La candela ha bisogno di due indicatori: uno per mostrare se è visibile, l'altro per mostrare se è accesa. L'indicatore F(0), non utilizzato, viene allora usato per "accendere" la candela. Se vi serve un indicatore in più per qualche cosa, potete utilizzare le caselle relative a parole che non ne usano, ad esempio la parola "NORD" (casella 19).

Dati per la matrice di oggetti posseduti

All'inizio del gioco il giocatore non ha nulla con sé; per indicare questo la matrice C() è vuota. Quando un oggetto viene raccolto, il computer scrive l nella sua casella. Quindi non servono istruzioni DATA per la matrice C().

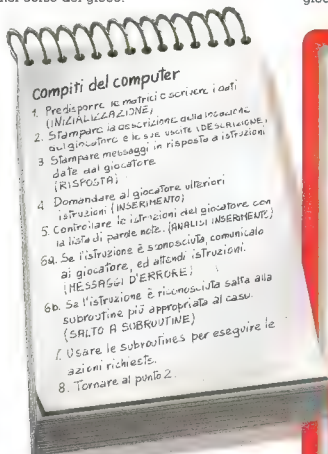
Inizializzazione

Predisporre le matrici e riempire i dati si dice "inizializzazione". Nel prossimo paragrafo vedrete dove sistemare questa operazione nella struttura del programma.

La struttura del programma

Per arrivare alla struttura globale del programma per il gioco d'avventura, dovete ancora considerare quali compiti dovranno essere eseguiti dal computer, nel corso del gioco.

pratica appare come segue. Come potete vedere, la parte maggiore è costituita dalle subroutine. Infatti ci dev'essere una subroutine per ogni verbo usato nel gioco. Altre informazioni a pagina 25.



La lista precedente mostra l'ordine in cui il computer deve eseguire le varie operazioni, che non è necessariamente l'ordine in cui devono apparire nel programma. Una buona parte del programma è costituito dalla routine di inizializzazione che serve una volta sola per ogni gioco, e benché questa sia la prima cosa che il computer deve fare, è buona idea metterla alla fine del programma. Questo perché, ogni volta che al computer è richiesto un GOTO o un GOSUB, questo torna all'inizio del programma e controlla ogni numero di linea finché non trova quello che gli serve. Ciò può portare via un bel po' di tempo, in un programma lungo. Ponendo la fase di inizializzazione alla fine, il computer non dovrà ricontrollarne i numeri di linea ad ogni mossa del giocatore. La struttura del programma in



Nell'interno del programma

Ora che avete un'idea generale su come si presenterà il programma, potete cominciare a pensare a ciascuna parte più in dettaglio. Avete già visto alle pagine 16 e 18 come funziona la fase di inizializzazione. Le 17 pagine seguenti descrivono il funzionamento delle altre parti principali del programma.

Descrizione e risposta

Ad ogni mossa, il computer deve dire al giocatore dove questi si trovi, e in quali direzioni possa andare. Deve anche far sapere al giocatore quello che è successo in conseguenza delle sue ultime istruzioni. Questa è la fase della descrizione e risposta, e funziona più o meno così. Provate ad identificare ciascuna parte sul listato del programma alle pagine ...

```

90 PRINT "TITOLO DEL GIOCO"
100 PRINT "
110 PRINT "LA TUA LOCAZIONE"
120 PRINT D$(RM)
  
```

RM è il numero della locazione in cui si trova il giocatore. Dovrete ricordarvi di dare un valore iniziale a questa variabile, nella fase di inizializzazione. (Per "La Casa Stregata", il valore iniziale per RM è 57, vedere la linea 2090).

Sentiero delle porte di ferro

Il computer cerca in D\$ (la matrice con tutte le descrizioni) ■ stampa ciò che trova nella casella numero RM.

Cosa c'è in D\$(57)?

```

130 PRINT "USCITE:"
140 FOR I=1 TO LEN (R$(RM))
150 PRINT MID$(R$(RM),I,1);";":
160 NEXT I
  
```

Questa routine esamina la lunghezza della stringa di caratteri nella casella numero RM della matrice dei percorsi, R\$. Il computer quindi cicla per un numero di volta pari a tale valore, stampando uno per volta i caratteri del contenuto di R\$(RM), ponendo tra ciascuno una virgola e lo spazio bianco.

"NOE" - così LEN\$(57) E' 3.

Le istruzioni 180-200 costituiscono un ciclo per vedere se nella locazione c'è un oggetto il cui indicatore è nullo (ovvero, se l'oggetto è visibile), e stamparne il nome se c'è.

```

180 FOR I=1 TO 0
190 IF L(I)=RM AND F(I)=0 THEN PRINT "QUI PUOI TROVARE";O$(I)
200 NEXT I

220 PRINT M$

225 M$="COSA ?"
  
```

M\$ è una variabile per contenere eventuali messaggi del computer al giocatore, in risposta a istruzioni date nel corso della mossa precedente. Cercate M\$ nel listato del programma da pag. 33 a pag. 37, per vedere come diverse risposte vengano inserite in M\$ in funzione di ciò che il giocatore aveva inserito da tastiera.

All'inizio di ciascuna mossa il contenuto di M\$ è inizializzato a: "COSA?" per cui se ciò non viene sostituito da altri messaggi, il computer si limita a stampare "COSA?", richiedendo la successiva azione del giocatore.

Messaggio urgente per il giocatore dalle subroutines.

CHE COSA?

M\$

Bene - Togliero "CHE COSA?"

La sezione d'inserimento

Un'importante caratteristica dei giochi di avventura consiste nel modo in cui il computer risponde ad istruzioni inserite da tastiera. "La Casa Stregata", come molti altri giochi, limita il giocatore all'uso di frasi di due parole, oltre all'uso di speciali istruzioni di una parola sola, come "AIUTO". La prossima sezione del programma dovrà chiedere istruzioni al giocatore, e quindi comunicare al computer cosa fare.

Per cominciare, il computer dovrà dividere la frase inserita dal giocatore in due parole, che potrà poi usare per un confronto con quelle che ha in memoria. La routine di divisione in parole utilizzata nella "Casa Stregata" funziona scandendo la frase inserita dal giocatore, finché non viene trovato uno spazio bianco tra le lettere. Tale routine viene listata qui, con l'aggiunta di poche linee per darvi modo di inserirla anche da sola, per vedere come funziona(*).

```
100 CLS
110 PRINT "BATTERE QUALCHE COSA"
120 INPUT Q$
130 V$=""
140 W$=""
150 FOR I=1 TO LEN(Q$)
```

Richiedono l'istruzione al giocatore, la ricevono e la scrivono in Q\$. Vengono poi predisposte due nuove variabili stringa: V\$ e W\$.

Serve a vedere di quanti caratteri è composta Q* e ad iniziare un ciclo per tale numero di volte.

```
160 IF MID$(Q$,I,1)="" AND V$="" THEN V$=LEFT$(Q$,I-1)
```

Questa linea cerca in Q\$ uno spazio bianco. Se ne trova uno, e V\$ è ancora vuota, scrive in V\$ tutte le lettere a sinistra dello spazio.



Continua a cercare in Q\$ finché non trova una lettera dopo uno spazio. Prende quindi tutto quel che si trova a destra dello spazio e lo scrive in W\$ (questo vuol dire che il numero di spazi che il giocatore pone tra le parole non ha nessuna importanza).

```
170 IF MID$(Q$,I+1,1)<>" " AND V$<>" " THEN W$=MID$(Q$,I+1, LEN(Q$)-1):I=LEN(Q$)
```

Quando V\$ e W\$ sono entrambe riempite, il contatore è posto al valore massimo, per uscire dal ciclo.

```
180 NEXT I
190 IF W$="" THEN V$=Q$
```

Se il computer non ha trovato spazi tra le lettere, allora V\$ e W\$ saranno ancora vuote alla fine del ciclo. Allora il computer prende l'intero contenuto di Q\$ e lo scrive in V\$.

```
200 M$="QUESTE SONO LE TUE DUE PAROLE"
210 PRINT "PRIMA PAROLA=";V$
220 PRINT "SECONDA PAROLA=";W$
230 IF W$="" THEN M$="HAI BATTUTO UNA SOLA PAROLA"
240 IF W$="" AND V$="" THEN M$="NON HAI BATTUTO NULLA"
250 PRINT M$
260 STOP
```

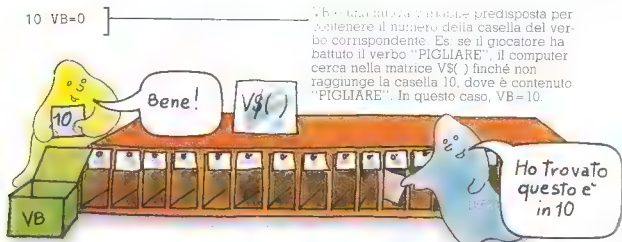
Questa sezione serve a far girare da sola la routine che divide le parole. Vengono quindi stampati messaggi in funzione di quello che avete inserito. Fatela girare per vedere cosa succede.

(*) N.B. Questa routine non funzionerà sui computers Sinclair (Timex). Vedere alle pagine 3* e 39.

L'analisi dell'inserimento

Il computer ora ha in memoria le istruzioni del giocatore contenute in due stringhe di caratteri V\$(stringa dei verbi) e W\$(stringa di parole). Compito successivo sarà di controllare tali istruzioni confrontandole con le parole che gli avete dato nella procedura di a pag. 18. Il computer presuppone che la parola contenuta in V\$ sia un verbo, e compie una ricerca per cicli successivi, per far corrispondere tale parola con uno dei verbi contenuti nella matrice V\$(). (Notate la differenza tra la variabile stringa V\$ e la matrice V\$() per il computer sono due cose diverse: cercate di non confonderle!).

Il computer quindi procede allo stesso modo per corrispondere il contenuto di W\$ con una delle parole contenute nella matrice O\$(). Ecco la sezione di programma che controlla la corrispondenza tra le parole inserite e le parole nella memoria del computer.



Vb è una nuova variabile predisposta per contenere il numero della casella del verbo corrispondente. Es. se il giocatore ha battuto il verbo "PIGLIARE", il computer cerca nella matrice V\$() finché non raggiunge la casella 10, dove è contenuto "PIGLIARE". In questo caso, VB=10.

```
20 FOR I=1 TO V
30 IF V$=V$(I) THEN VB=I
40 NEXT I
```

Il computer cicla per V volte (V = n° verbi in memoria) confrontando il verbo dato dal giocatore con quelli in memoria. Se trova quello che corrisponde, attribuisce a VB il valore opportuno.

```
50 OB=0
60 FOR I=1 TO W
70 IF W$=O$(I) THEN OB=I
80 NEXT I
```

Il ciclo per W\$ funziona allo stesso modo, utilizzando OB per registrare il numero di casella della parola corrispondente a quella data.

Se le parole non corrispondono?

Nel caso non vi siano corrispondenze, VB e OB saranno entrambe uguali a 0. Il computer interpreta ciò come se la parola corrispondente a quella fornita si trovasse nella casella zero. Ma quando va a cercare in tale casella la parola corrispondente, non vi trova nulla, perché l'avevate lasciata vuota quando avevate scritto i dati nelle matrici.



Combinazioni prive di senso

Attenzione, questo processo di ricerca della corrispondenza si limita a controllare che le due parole date siano in memoria, ma non controlla che la combinazione delle due parole abbia senso. Una combinazione priva di senso, come: DISSERRARE LA CANDELA, viene accettata in questa fase del programma, ma verrà rifiutata successivamente quando il computer proverà ad eseguire l'azione richiesta. A questo punto del programma è molto più rapido limitarsi a controllare le parole separatamente, piuttosto che dire al computer di controllare la validità delle combinazioni. Al termine di questa sezione del programma il computer ha un valore per VB ed uno per OB.

Predisporre i messaggi d'errore

Il computer può usare i valori di VB e OB per controllare se deve mandare al giocatore un messaggio di istruzioni non valide. Questa parte del programma funziona da filtro. Le istruzioni fornite dal giocatore sono sottoposte ad alcuni test. Se non superano anche uno solo dei test, viene posto un nuovo messaggio in M\$. Se invece superano tutti i test, M\$ conterrà ancora il messaggio "COSA?", scritto inizialmente alla linea 220. (In questa fase i messaggi sono predisposti in memoria, ma non vengono stampati sullo schermo, possono essere cambiati, nel corso del programma). Ecco le linee di programma che predispongono i messaggi di errore della "Casa Stregata" cercate di individuarli nel listato principale. Se scrivete da soli un'avventura dovrete includere anche linee di questo tipo.



Ricorda che il messaggio in M\$ potrà essere modificato nel corso del programma.

1

```
IF W$="" AND OB=0 THEN M$="CHE  
SCIOCCHENZA"
```

Che sciocchezza!

SPRUZZARE
VERNICE

Il primo test controlla che vi sia una parola in W\$ (ovvero che il giocatore abbia inserito due parole) e quindi controlla se il valore di OB sia nullo.

2

```
IF VB=0 THEN M$="V"
```

Questa linea serve a superare un problema. Non potete usare l'istruzione GOSUB con valore zero. I diversi tipi di BASIC variano nel tipo di risposta. La maggior parte di essi ignora il GOSUB e continua con la linea successiva. Altri invece, come il BBC, fanno obiezioni allo zero, ed emettono un messaggio di "errore di intervallo". Per superare

questa difficoltà, VB viene cambiato da zero ad un valore maggiore di V (il numero di verbi nella memoria del computer) e il computer viene inviato ad una subroutine fittizia.

3

```
IF M$="" THEN M$="MI SERVONO  
DUE PAROLE"
```

COLPIRE

Mi servono due parole

Questa linea prepara un messaggio nel caso in cui il giocatore abbia inserito solo una parola, e W\$ sia vuota. (Se la parola fa parte di quei comandi di una sola parola permessi al giocatore, questo messaggio verrà modificato più avanti).

4

```
IF V$="" AND OB=0 THEN M$="NON  
PUOI " + Q$ + ""
```

UCCIDERE I
FANTASMI

Non puoi
"Uccidera i fantasmi"

Questa linea dispone un messaggio se il computer non trova il verbo dato fra quelli in memoria, ma trova il nome dell'oggetto.

5

```
IF VB>V AND OB=0 THEN M$="QUESTO  
NON HA SENSO"
```

ARREN-
DERSI

Questo non ha senso!

Se il computer non trova in memoria nessuna delle due parole fornite dal giocatore, predisporre questo messaggio.

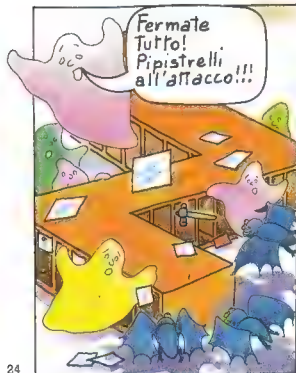
```
IF VB<4 AND OD=0 AND C(OB)<3 THEN
  M="NOX 'KAT' '03'03"
```

ACCENDERE
LA CANDELANon hai
le candele

Condizioni di annullamento

A volte nel corso di un'avventura avvengono cose che impediscono al giocatore di fare alcunché, prima di averle affrontate. In queste circostanze, delle istruzioni che normalmente sarebbero considerate valide, devono essere annullate, per cui il computer ha bisogno di alcune linee di programma che predispongono degli indicatori in memoria per comunicargli la validità di certe condizioni. (*)

Nella "Casa Stregata" le linee da 420 a 450 costituiscono delle condizioni di annullamento. Potete vedere qui di seguito tali linee, con una breve descrizione del loro funzionamento.



```
420 IF F(26)=1 AND RM=13 AND RND(3)<3
  AND VB<21 THEN M$="PIPISTRELLI
  ALL'ATTACCO":GOTO 90
430 IF RM=44 AND RND(2)=1 AND F(24)<1
  THEN F(27)=1
440 IF F(0)=1 THEN LL=LL-1
450 IF LL<1 THEN F(0)=0
```

Linea 420

Se vi sono pipistrelli presenti, il giocatore si trova nella stanza della torre sul retro, il numero casuale non vale 3, e il giocatore non ha usato il verbo Z1 (SPRUZZARE) nelle sue istruzioni, allora M\$ è posto uguale a "PIPISTRELLI ALL'ATTACCO" e il giocatore non può proseguire il gioco.

Linea 430

Se il giocatore è nella stanza piena di ragnatele, il valore del numero casuale non è 1, e l'aspirapolvere è spento allora viene attivato un indicatore per far apparire i fantasmi paralizzanti. Ovvero, l'indicatore F(27) è posto uguale a 1.

Linea 440

Se la candela è accesa, allora il contatore limitatore di luce LL viene diminuito di 1.

Linea 450

Se LL vale zero, allora l'indicatore di candela accesa/spenta F(0) viene posto uguale a zero.

Potrete forse anche voi pensare ad altre condizioni di annullamento, da aggiungere a queste.

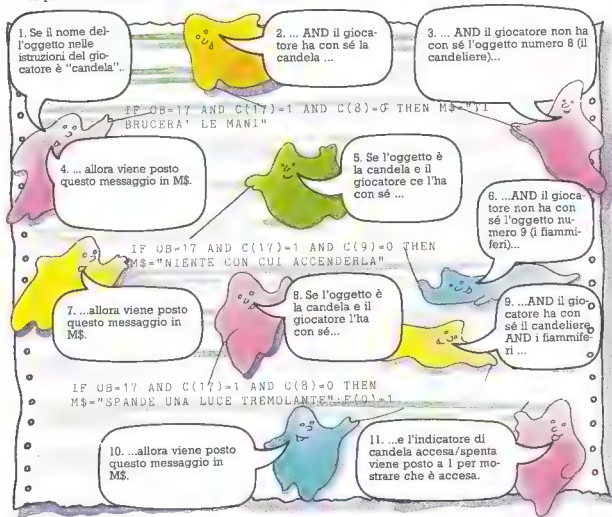
Saltare alle subroutines

Il successivo compito del computer è quello di tentare di eseguire le azioni che il giocatore voglia fare. Se ogni volta il computer dovesse cercare nella lista delle azioni possibili fino a trovare quella richiesta dal giocatore, il gioco sarebbe molto lento e noioso. Per evitare che questo accada, si utilizzano molte subroutines — una per quasi ogni verbo PRENDERE possono usare la stessa subroutine). Potrete allora usare un'istruzione di ON ... GOSUB per dire al computer di saltare a diverse subroutines in funzione del valore di VB.

```
ON VB GOSUB 500,570,640,640,640,
570,640,640,640,640,640,640,640,
1140,1180,1220,1250,1300,1340,
1380,1400,1430,1460,1490,1510,1590
```

Come funziona la linea ON ... GOSUB

La linea ON ... GOSUB appena vista funziona così. Se VB=1 il computer salta all'istruzione del primo numero di linea listato (500). Se VB=2, va alla seconda, se VB=3 alla terza e così via. Notate che l'ultimo numero di linea listato è quello di una subroutine fittizia, per VB=V+1 (cioè il valore assegnato a VB se non viene trovato nella memoria nessun verbo corrispondente). La linea alla quale il computer viene mandato in questo caso, contiene solo l'istruzione RETURN, e quindi rimanda direttamente all'esecuzione del programma principale. Guardate le subroutines alle pagine 34 e 36, e cercate di capire a che cosa servono. Troverete alle linee 1340-1370 la procedura relativa al verbo ACCENDERE (VB=19).



Che succede se l'oggetto non è la candela?

Se l'oggetto che il giocatore volesse usare non fosse "LA CANDELA" ma uno degli altri contenuti della memoria del computer, come "LA PORTA", il messaggio contenuto in M\$ resterebbe invariato rispetto a quello predisposto alla linea 220. Quando il computer ritorna al programma principale e trova l'istruzione: PRINT M\$, stamperà il messaggio "COSA?". Non c'è bisogno di prevedere un messaggio che dica che la candela non c'è, perché questo è già previsto nella sezione dei messaggi d'errore.

Ritorno al programma principale

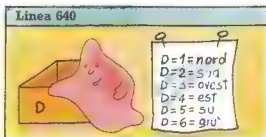
Benché alcune delle routines dei verbi siano più lunghe e più complicate di questa, funzionano tutte allo stesso modo: viene controllato il valore di OB, è preparato se necessario un messaggio speciale, poi il computer ritorna al programma principale. Quindi viene controllato il limite di durata della luce alle linee 470 e 480, poi l'esecuzione passa alla sezione di descrizione e risposta. Qui viene stampato il messaggio che era stato posto in M\$. Ora il computer resta in attesa delle successive istruzioni.

La subroutine "ANDARE"

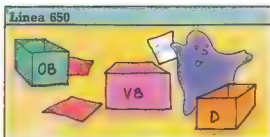
La subroutine per il verbo ANDARE è così importante in un gioco d'avventura, che la potreste considerare come un programma nel programma. Essa viene richiamata da sette istruzioni contenenti sette verbi diversi — ANDARE, N, S, O, E, A e B. Questa routine è speciale anche perché risponde a comandi direzionali forniti con un'unica lettera, oltre a quelli di due parole. Non è obbligatorio includere questa possibilità nel vostro programma, ma essa aiuta a rendere il gioco più veloce e più interessante da giocare. Se avete già giocato con le avventure, saprete quanto sia noioso inserire tutte le volte: ANDARE A NORD; etc.

Ecco come funziona la routine ANDARE:

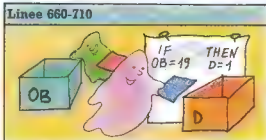
Ecco come funziona la routine ANDARE.



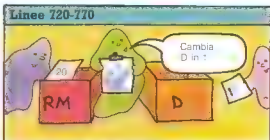
All'inizio viene creata una variabile D per contenere l'informazione sulla direzione in cui il giocatore si vuol muovere. I suoi valori, da 1 a 6, corrispondono a: nord, sud, ovest, est, in alto ■ in basso.



La linea successiva controlla se il giocatore abbia inserito solo una parola, e poi attribuisce a D un valore in funzione del valore di VB. (Notare che sottraendo 3 a VB, il computer ricava valori per D che corrispondono a quelli delle linee 660-710).



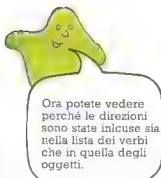
Le sei linee successive controllano se il giocatore abbia inserito un'istruzione di spostamento composta da due parole. Queste linee usano il valore di OB per dare un valore a D.



Poiché questa non è propriamente un'avventura in tre dimensioni, le istruzioni IN ALTO e IN BASSO devono essere convertite in nord, sud, est o ovest. Questo viene fatto dalle linee 720-770. (Se vi riferite allo schema principale alla pagina 26 vedrete che funziona).

```

640 D=0
650 IF OB=OTHERD=VB-3
660 IF OB=19 THEN D=1
670 IF OB=20 THEN D=2
680 IF OB=21 THEN D=3
690 IF OB=22 THEN D=4
700 IF OB=23 THEN D=5
710 IF OB=24 THEN D=6
720 IF RM=2D AND D=OTHERD=1
730 IF RM=2D AND D=OTHERD=3
740 IF RM=2D AND D=OTHERD=5
750 IF RM=2D AND D=OTHERD=7
760 IF RM=36 AND D=OTHERD=1
770 IF RM=36 AND D=OTHERD=2
780 IFF(14)=1 THEN MS="CRASH! SEI CADUTO D
ALL'ALBERO":F(14)=0:RETURN
790 IFF(27)=1 AND RM=52 THEN MS="I FANTASMI
TI IMPEDISCONO DI MUOVERTI":RETURN
800 IF RM=45 AND C(1)=1 AND F(34)=0 THEN MS="UN
A BARRIERA MAGICA AD OVEST":RETURN
810 IF (RM=26 AND F(0)=0) AND (D=1 OR D=4) THEN M
S="PER ANDARE DI LA' DEVI FARE LUCE":RET
URN
820 IF RM=5R AND C(15)<1 THEN MS="SEI BLOCCA
TO!":RETURN
830 IF C(15)=1 AND NOT (RM=53 OR RM=54 OR RM=55)
R M<47) THEN GOTO 835
833 GOTO 840
835 M3="NON PUOI TRASPORTARE UNA BARCA!":
RETURN
840 IF (RM=26 AND M<30) AND F(0)=0 THEN MS="TR
OPPO BUIG PER MUOVENSI":RETURN
    
```





Il computer ha bisogno di controllare l'esistenza di condizioni speciali con conseguenze sulle possibilità di movimento del giocatore. Es.: se $F(14)=1$, il giocatore si trova sull'albero. Se cerca di muoversi senza scendere dall'albero, riceverà un messaggio che gli dirà che è caduto.

Se il giocatore si trova nella locazione 52 e l'indicatore per i fantasmi è posto a 1, gli viene inviato un messaggio per dirgli che non si può muovere. Ciascuna di queste condizioni fa tornare il computer al programma principale. Provate a capire a cosa servono le altre linee.

Rilevare la presenza dei muri

Se la mossa non è stata bloccata da una di queste condizioni speciali, il computer dovrà verificare ancora che non vi sia un muro, o altro, ad impedire il cammino. Ecco le linee di programma che svolgono questa funzione. A prima vista sembrano abbastanza complicate, ma se esaminate con cura ciascuna parte, ricordando cosa rappresentano tutte le variabili, dovrete poterle capire il funzionamento.

Indicatore che il computer utilizza per registrare se ha trovato l'uscita richiesta dal giocatore.

RL è una variabile che contiene la lunghezza della stringa di caratteri trovati in R(RM)$. (Questa rappresenta i percorsi, N, S, O, etc., che partono dalla locazione in cui il giocatore si trova).

Il computer cicla per RL volte.

Ad ogni ciclo il computer preleva uno dei caratteri di R(RM)$ e lo chiama temporaneamente U\$.

Quindi il computer effettua una serie di tests su U\$ e D. Se l'istruzione di spostamento del giocatore corrisponde ad una uscita della locazione in cui si trova. Il valore di RM viene cambiato per "muovere" il giocatore alla locazione opportuna. F(35) viene poi posto uguale a 1 perché il computer non cambi di nuovo il valore di RM, o effettui un'altra serie di cicli. Ciò potrebbe essere possibile, in quanto il computer utilizza il nuovo valore di RM nella linea 870).

Se controllate lo schema funzionale, vedrete come l'aggiunta o la sottrazione di 1 o 8 muova il giocatore alla giusta locazione successiva.

Alla fine del ciclo, in M\$ viene scritto "OK". Questo sostituirà il messaggio: "MI SERVONO DUE PAROLE", incluso tra i messaggi di errore, se il giocatore ha inserito uno spostamento con una parola sola.

Se F(35) è ancora zero, la direzione in cui il giocatore vuole andare non è permessa, e M\$ viene cambiato per farlo sapere al giocatore.

Se $D < 1$, (non gli era stato assegnato un valore alle linee 650-770) M\$ viene cambiato in "ANDARE DOVE?".

Questa linea rende la porta d'ingresso a percorso a senso unico. Quando il giocatore entra nella locazione 41 (l'atrio), le uscite dalla locazione 49 (portico d'ingresso) vengono cambiate da "NSW" a "SW", in M\$ viene scritto "LA PORTA SI CHIUDE DI COLPO!", e l'indicatore per la porta d'ingresso viene posto a zero per indicare che la porta è chiusa. (I percorsi dalla locazione 41 non devono essere cambiati).

```

850 F(35)=0:RL=LEN(R$(RM))
860 FORI=1TORL
870 U$=MID$(R$(RM),I,1)
880 IF(U$="N"ANDD=1ANDF(35)=0)THENRM=RM-8:F(35)=1
890 IF(U$="S"ANDD=2ANDF(35)=0)THENRM=RM+8:F(35)=1
900 IF(U$="O"ANDD=3ANDF(35)=0)THENRM=RM-1:F(35)=1
910 IF(U$="E"ANDD=4ANDF(35)=0)THENRM=RM+1:F(35)=1
920 NEXTI
930 M$="OK"
940 IFF(35)=0THENM$="NON PUOI ANDARE DA QUELLA PARTE!"

```

```

950 IFD<1THENM$="ANDARE DOVE?"
960 IFRM=41ANDF(23)=1THENR$(49)="SO":M$="LA PORTA SI CHIUDE DI COLPO!":F(23)=0
970 RETURN

```

Cambiare il programma

Potete cambiare il programma contenuto in questo libro quanto piú vi piace, sia per produrre variazioni sul tema della "Casa Stregata", sia per creare giochi con ambienti, descrizioni, oggetti, verbi ■ messaggi completamente diversi. Ricordatevi che tanto piú cambiate, tanto piú complicato sarà il lavoro, poiché dovrete pensare a come ciascun cambiamento influenzi il resto del gioco. Se decidete di scrivere un gioco nuovo, utilizzando questo programma come guida, allora dovrete pianificarlo come descritto alle pag. da 6 ■ 15. Vale la pena di dedicare un po' di tempo a pianificare correttamente il gioco, così avrete minor probabilità di trovarlo pieno di errori quando farete girare il programma. È buona idea cominciare facendo piccoli cambiamenti, all'inizio, per vedere cosa succede. Se salvate il programma principale su nastro, potrete effettuare cambiamenti, provarli ed adattarli senza perdere l'originale.

Quanta memoria libera avete?



Il programma della "Casa Stregata" occupa circa 7K di RAM, prima di iniziare a girare. Poi ha bisogno di circa 3.5-4K di RAM per le matrici in cui immagazzinare i dati. Il computer utilizzerà anche un po' di memoria per le funzioni interne ed userà ancora 1K o piú per lo schermo. (Lo SPECTRUM, per le sue funzioni, necessita di 7K ecco perché "La Casa Stregata" non può girare sul modello a 16K). Perciò, se avete un computer ■ 16K, con poca memoria di scorta, la maggior parte dei cambiamenti dovranno essere delle sostituzioni, non delle aggiunte.

Descrizioni piú lunghe

Se avete oltre di 16K di memoria, uno dei modi piú semplici per rendere il gioco interessante è quello di aggiungere descrizioni piú lunghe. Al posto di "impressionante sala con soffitto ■ volta", ad esempio, potrete avere qualcosa del tipo "Siete entrati in un'ampia stanza, con soffitto a volta e colonne che si allungano sopra di voi, di molte volte la vostra altezza. La luce filtra nella stanza da est, e sembra esservi una porta lontana, ad ovest ...".



Aggiunta di un limite di tempo

"La Casa Stregata" ha già un limite di tempo imposto dalla durata della candela. Potreste aggiungere anche un tempo limite globale al gioco, facendo contare al computer il numero di mosse fatte dal giocatore, e facendo interrompere il gioco ad un numero prefissato. Questo vi sarà possibile facendo un'aggiunta alla linea 70, e inserendo una nuova linea 485, come segue:

```
70 V=25;W=36;G=18;T=0
```



Potreste anche non dover fare così. Non tutti i computers devono inizializzare le variabili prima di usarle.

```
485 T=T+1:IF T>200 THEN PRINT  
"E' MEZZANOTTE: TI SEI  
TRASFORMATO IN FANTASMA":STOP
```

Potreste cambiare questo numero in qualsiasi altro, a vostra scelta.

Quesito

Riuscite a pensare a come stabilire un limite al numero di oggetti che possono essere trasportati contemporaneamente dal giocatore? (Vi servirà anche modificare la routine di calcolo del punteggio).

Aggiunta di suoni



Ecco un modo per arricchire il gioco senza fare troppi cambiamenti. Dovete conoscere il funzionamento delle istruzioni per il suono del vostro

computer. Fate delle prove con le routine sonore, per assicurarvi che siano come le volete voi, poi aggiungete un'istruzione GOSUB, di salto alla linea dove avviene l'azione; esempio: IF RM=46 AND C(1)=1 THEN M\$="STA SUCCEDENDO QUALCOSA DI SPAVENTOSO": GOSUB 6000.

Potreste aggiungere suoni per la porta che sbatte, il muro segreto che crolla, la chiave che gira nella serratura, e così via. La tabella che segue fornisce alcune routines di suono per diversi computer. Aggiungetene quante ne consente la memoria.

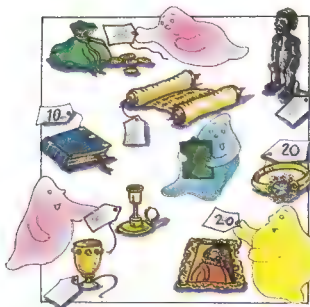
	CADUTA DELL'ALBERO	PORTA CHE SBATTE	MAGIA	COLPI D'ASCIA	RICHIAMO GENERALE
VIC 20	POKE 36877,130 FOR L=15 TO 0 STEP -1 POKE 36878,L FOR M=1 TO 20:NEXT M NEXT L POKE 36877,0	POKE 36877,130 FOR L=15 TO 8 STEP -1 POKE 36878,L FOR M=1 TO 2:NEXT M NEXT L POKE 36877,0	POKE 36878,15 FOR I=150 TO 240 STEP 8 POKE 36876,I FOR M=1 TO 100:NEXT M POKE 36876,0	POKE 36878,15 FOR I=1 TO 10 POKE 36877,200 POKE 36877,0 FOR M=1 TO 400:NEXT M NEXT I	POKE 36878,15 FOR I=1 TO 2 POKE 36876,200 POKE 36876,0 FOR M=1 TO 400:NEXT M NEXT I
SPECTRUM	FOR L=15 TO 0 SOUND 0,L,5,1, NEXT L	—	FOR I=5 TO 40 STEP 3 BEEP 0,0,1 NEXT I	FOR I=1 TO 10 BEEP 0,01,0,01 FOR M=1 TO 100 NEXT I	BEEP 0,5,5 PAUSE 50 BEEP 0,5,5
BBC	FOR L=0 TO 15 SOUND 0,L,5,1 NEXT L	FOR L=15 TO 8 SOUND 0,L,5,0,6 NEXT L	FOR I=40 TO 160 STEP 8 SOUND 2,-15,1,5 NEXT I	FOR I=1 TO 10 SOUND 0,-15,5,1 FOR M=1 TO 1000:NEXT M NEXT I	SOUND 2,-15,100,2 FOR M=1 TO 1000:NEXT M SOUND 2,-15,100,2
	—	—	—	—	—
C 64	POKE 54273,20 POKE 54278,255 POKE 54276,129 FOR L=15 TO 8 STEP -1 POKE 54295,L FOR M=1 TO 10 : NEXT M POKE 54273,0 POKE 54278,0 POKE 54275,0	POKE54273,20 POKE54278,255 POKE54276,129 FOR L=15 TO 0 STEP -1 POKE 54295,L FOR M=1 TO 2:NEXT M NEXT L POKE 54273,0 POKE 54278,0 POKE 54276,0	POKE54276,17 POKE 54278,255 POKE 54295,15 FOR I=40 TO 120 STEP 8 POKE 54273,I FOR M=1 TO 100:NEXT M NEXT I POKE 54273,0 POKE 54278,0 POKE 54276,0	POKE 54278,255 POKE 54276,129 POKE 54295,15 FOR I=1 TO 10 POKE 54273,20 POKE 54273,0 FOR M=1 TO 600:NEXT M NEXT I POKE 54276,0 POKE 54278,0 POKE 54296,0	POKE 54278,255 POKE 54276,129 POKE 54295,15 FOR I=1 TO 2 POKE 54273,20 POKE 54273,0 FOR M=1 TO 600:NEXT M NEXT I POKE 54276,0 POKE 54278,0 POKE 54296,0

Conquistare punti

"La Casa Stregata" ha un sistema di conquista dei punti molto semplice: un punto per ogni oggetto in possesso del giocatore. Potrete cambiare questo sistema con uno piú interessante, come il basare il punteggio sul valore dell'oggetto. Pensate che gli oggetti siano numerati in ordine decrescente di valore, il quadro sarà quello di maggior valore, e la chiave quello di valore minore. Se modificate la linea 1530 così:

```
1530 IF C(I)=1 THEN S=S+G-1
```

allora il quadro avrà il valore $18-1=17$, e la chiave $18-18=0$ (G è il numero di oggetti che il giocatore può trasportare e I è il numero dell'oggetto che possiede). Ciò rende la chiave priva di valore in quanto parte del tesoro, ma di grande valore in quanto oggetto utile: senza di essa non si riuscirebbe ad arrivare al quadro o al calice.



Se voleste un sistema piú flessibile (purché abbiate abbastanza memoria nel computer), potreste costruire una matrice contenente i valori degli oggetti nella routine di inizializzazione, in questo modo:



Usate i valori che preferite.

Aggiungete i numeri di linea.

```
DIM T(G)
FOR I = 1 TO G
READ T(I)
NEXT I
```

Dimensionate la nuova matrice T con G caselle (tanti quanti sono gli oggetti trasportabili).

Modificate anche la linea 1530 come segue:

```
1530 IF C(I)=1 THEN S=S+T(I)
```

che sciocchezza

Penalità

Fino ad ora, la routine del punteggio del giocatore contava solo punti da aggiungere e non era influenzata da altre azioni del giocatore.

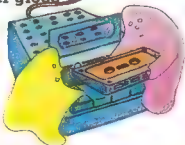
Potreste facilmente aggiungere un sistema di penalità usando un contatore, MK, degli errori. Ad ogni errore del giocatore, incrementate MK, e alla fine ne sottraete il valore da S, quando il punteggio viene calcolato alla linea 1530. Ad esempio se il giocatore cade dall'albero, attribuitegli un punto di penalità, in questo modo:

```
780 IF F(14)=1 THEN M$="CRASH SEI  
CADUTO":F(14)=0:MK=MK+1:  
RETURN
```

Non dimenticate che certi computers hanno bisogno di definire le nuove variabili prima di poterle usare. Potete fare questo aggiungendo $MK=0$ alle variabili della linea 70.

Salvataggio del gioco

Potrebbe essere comodo poter interrompere un gioco a metà strada, per poi riprenderlo piú tardi nel punto dove l'avete lasciato. Con i giochi lunghi e complicati, questa è una caratteristica molto importante, e la potete includere aggiungendo le voci SALVARE e CARICARE, alla lista dei verbi. Alla linea 70, cambiate il valore di V ed aggiungete i due nuovi verbi, separati dalla virgola, alla fine della linea 1665. Dovrete anche cambiare la linea ON GOSUB alla linea 460.



Ponete i numeri di linea delle due nuove subroutines (uno per SALVARE, l'altro per CARICARE) tra gli ultimi due numeri alla linea 460, cosicché questa diventi:

... 1510, 3000, 4000, 1590

Primo nuovo numero Secondo nuovo numero

Poi aggiungete le due nuove subroutines così, controllando il manuale del vostro computer per accertarvi che le istruzioni siano corrette.

```
*3000 INPUT "SIETE PRONTI A
REGISTRARE";I$
3010 IF I$<>"S" THEN GOTO 3000
3020 APRIME IL FILE IN
USCITA DAL COMPUTER
3030 PRINT #1,RM
```

Sostituite questa linea con le istruzioni proprie del vostro computer. Potreste anche non aver bisogno di nessuna linea qui.

Questa linea salva il numero della locazione in cui si trova il giocatore.

Questo ciclo salva la posizione degli oggetti trasportabili.

```
3040 FOR I=1 TO G
3050 PRINT #1,L(I)
3060 NEXT I
3070 FOR I=1 TO W
3080 PRINT #1,C(I),F(I)
3090 NEXT I
3100 CLOSE
3200 RETURN
```

Questo salva gli oggetti in possesso del giocatore, e lo stato degli indicatori.

```
*4000 INPUT "SIETE PRONTI A
CARICARE";I$
4010 IF I$<>"S" THEN GOTO 4000
4020 APRIME IL FILE IN
ENTRATA AL COMPUTER
4030 INPUT #1,RM
4040 FOR I=1 TO U
4050 INPUT #1,L(I)
4060 NEXT I
4070 FOR I=1 TO W
4080 INPUT #1,C(I),F(I)
4090 NEXT I
4100 CLOSE
4200 RETURN
```

Notate che questa routine di salvataggio non salva le descrizioni ed i percorsi del gioco. Questo vuol dire che le stanze e i percorsi alterati dalle azioni del giocatore ritorneranno al loro stato originale — il muro segreto sarà ricostruito, la porta sarà di nuovo sprangata, e così via. (Potete dar la colpa ai fantasmi). Potreste salvare le matrici D\$ e R\$, volendo, con l'aggiunta di altri cicli alle routine SALVARE e CARICARE.

* Se avete un modello BBC, potreste dover sostituire il punto e virgola con la virgola.

Vi arrendete?

Come la maggior parte delle avventure, "La Casa Stregata" contiene trappole per il giocatore, e queste possono essere evitate solo utilizzando un certo oggetto in un certo modo. Se il giocatore non possiede quell'oggetto, resta bloccato. Una funzione di "abbandono" può essere utile in questa situazione, così che il giocatore non debba premere il tasto BREAK o ESCAPE per terminare il gioco. Potrete realizzare tale funzione aggiungendo ABBANDONARE alla lista dei verbi, ed inserendo una nuova subroutine, come per SALVARE e CARICARE descritte prima.



Dovete ricordarvi di cambiare il valore di V alla linea 70, di aggiungere ABBANDONARE alla fine della linea 1665, e di inserire il numero di linea della nuova subroutine nella linea 460, ponendolo in penultima posizione.

La subroutine ABBANDONARE dovrà essere più o meno così:

```
5010 IF Q$<>"S" THEN RETURN
5020 INPUT "VUOI PRIMA SALVARE IL
GIOCO";Q$
5030 IF Q$="S" THEN GOSUB 3000
```

Queste linee non vi servono se non avete previsto nel programma la funzione SALVARE.

```
5040 PRINT "GRAZIE PER AVER
GIOCATO!"
5050 END
```

Notate che non vi è nessuna istruzione di RETURN alla fine di questa subroutine. Questo in genere è contro le regole del BASIC, ma in questo caso il computer non rischia di confondersi, perché il programma avrà già smesso di girare, una volta che si arrivi alla linea 5050.

Il debugging della vostra avventura

Se scrivete la vostra versione della "Casa Stregata" o utilizzate le routines contenute in essa per creare una nuova avventura, allora è probabile che vi capiti di fare degli errori. Trovare gli errori ed eliminarli è un'operazione detta "debugging" (spulciamento). Ecco alcuni dei problemi che potrete incontrare ed alcuni suggerimenti per risolverli.

Troppo pochi dati

Se il computer vi fornisce un codice d'errore del tipo: "troppo pochi dati nella linea X", significa che le quantità non corrispondono, in una delle sezioni di lettura dati. Controllate che il numero di termini o valori dati sia uguale al numero di cicli usato per leggerli. Potreste aver dimenticato una virgola nell'istruzione di DATA, o un valore, oppure aver richiesto un numero di cicli sbagliato.

```
DIM A(4)
FOR I=1 TO 4
READ A(I)
NEXT I
DATA LA SPADA, LE MONETE, IL
CIBO L'ACQUA
```

Manca la virgola.

Errore di matrice

Una segnalazione di errore di matrice, vuol dire che non avete riservato abbastanza caselle con l'istruzione di DIMENSIONAMENTO della matrice, oppure avete inserito un termine in più nell'istruzione di DATA (forse una virgola di troppo), e quindi avete contato anche questo termine nel numero di cicli dell'istruzione READ.

```
DIM A(3)
FOR I=1 TO 4
READ A(I)
NEXT I
DATA L'ASCIA, LA BARA, IL
SANGUE, LA CHIAVE
```

Qui c'è un errore.

Oggetti che si comportano in modo strano

Questo può succedere poiché il programma viene mandato alla subroutine sbagliata, dalla linea ON GOSUB. Confrontate ciascun numero di tale linea con la subroutine che inizia a quel numero di linea. Se queste sono tutte giuste, controllate che vi sia una linea di RETURN alla fine di ciascuna subroutine. Se questa manca, il computer eseguirà le istruzioni "in cascata" con la subroutine successiva, il che può produrre degli strani risultati.

Se nessuna delle cose viste risolve il problema, allora verificate attentamente le condizioni presenti nella subroutine. Potreste aver dimenticato qualcosa, o messo un segno sbagliato, o aver usato la variabile errata per sbaglio. Controllare anche le condizioni di annullamento e gli indicatori usati precedentemente nel programma.

Uscite in posti strani

Se trovate un muro attraverso il quale potete camminare, o una porta aperta che non potete varcare, forse avete fatto un errore nel pianificare i percorsi o nell'inserire i dati di percorso. Confrontate la mappa dei percorsi con le linee contenenti tali dati.

Oggetti che non appaiono dove dovrebbero

Se un oggetto appare al posto sbagliato, avete fatto un errore nell'inserire i dati per la matrice I. Se un oggetto non appare affatto, controllate la matrice degli indicatori. Avete forse posto l'indicatore con il numero di quell'oggetto a 1, che vuol dire, che l'oggetto c'è, ma il computer non ve lo dice. Dovrete porre l'indicatore a zero. Controllate la routine di inizializzazione, ed anche i riferimenti agli indicatori nel programma.

Il listato della "Casa Stregata"

Ecco il listato del programma per l'avventura "La Casa Stregata". Dovrebbe poter girare su qualsiasi computer che usi un BASIC "tipo Microsoft", e che abbia un minimo di 16K di RAM. Potreste dover effettuare delle piccole modifiche per il vostro computer - attenti alle note riportate accanto ad alcune linee del programma. Se avete un BBC vi è una modifica da apportare, che troverete alla fine del listato a pag. 37. Se avete un BBC modello A, usate il modo 7. Questo listato non funzionerà così com'è sui computers Sinclair (Timex). Se avete uno Spectrum (Timex 2000) andate a pag. 38 per vedere le modifiche da apportare al programma. Se avete uno ZX81 (Timex 1000), c'è un listato speciale per voi alle pag. 39 a 45.

Poiché questo è un programma lungo, dovrete stare molto attenti quando lo inserite da tastiera. Il minimo errore potrebbe impedirgli di girare correttamente, e sarebbe molto difficile da trovare. Controllate subito ciascuna linea nel corso dell'inserimento, specialmente le linee di salto alle subroutine e DATA. Alcune linee di programma sono così lunghe che richiedono due o più linee di stampa. Fate attenzione a queste ultime, e accertatevi di non premere RETURN o ENTER fino alla fine di ciascuna linea.

Se avete un BBC, sostituite CLS a PRINT CHR\$(147). Se avete un APPLE, sostituitelo con HOME.

La linea 70 inizializza le variabili. V è il numero dei verbi, W è il numero di nomi di oggetti, G è il numero di oggetti "trasportabili".

Questa linea manda il programma alle routine di inizializzazione.

Vedere a pag. 20 per scoprire come funziona la sezione di descrizione e risposta.

Se avete un BBC, qui potreste aver bisogno di una virgola al posto del punto e virgola.

Vedere a pag. 21-22 per scoprire come funziona la sezione di inserimento.

```

10 REM L'AVVENTURA DELLA CASA STREGATA
20 REM *****
30 REM QUESTA VERSIONE PER BASIC COMMODORE
40 REM NECESSITA ALMENO DI 16K
60 REM *****
70 V=25:W=36:G=18
80 GOSUB1600
90 PRINT CHR$(147):PRINT"LA CASA STREGATA"
100 PRINT"-----"
110 PRINT"LA TUA LOCAZIONE"
120 PRINTD$(RM)
130 PRINT"USCITE:";
140 FORI=1TOLEN(R$(RM))
150 PRINTMID$(R$(RM),I,1);",";
160 NEXTI
170 PRINT
180 FORI=1TOG

```

```

190 IFL(I)=RMANDF(I)=0THENPRINT"QUI PUOI TROVARE ";O$(I)
200 NEXTI
210 PRINT"-----"
220 PRINTM$(M$)="COSA ?"
230 INPUT"COSA VUOI FARE ADESSO":Q$
240 V$="":W$="":VB=0:OB=0
250 FORI=1TOLEN(Q$)
260 IFMID$(Q$,I,1)=" "ANDV$=""THENV$=LEFT$(Q$,I-1)

270 IFMID$(Q$,I+1,1)<>" "ANDV$<>" "THENW$=MID$(Q$,I+1,LEN(Q$)-1):I=LEN(Q$)
280 NEXTI
290 IFW$=""THENV$=Q$
300 FORI=1TOV
310 IFV$=V$(I)THENVB=I
320 NEXTI
330 FORI=1TOW
340 IFW$=O$(I)THENOB=I
350 NEXTI

```

```
360 IFW$>"ANDOB=OTHENM$="CHE SCIOCCHEZZA"
370 IFVB=OTHENVB=V-1
380 IFW$="THENM$="MI SERVONO DUE PAROLE"
390 IFVB>VANDOB>OTHENM$="NON PUOI ""+Q$+""
400 IFVB>VANDOB=OTHENM$="QUESTO NON HA SENSO"
```

```
410 IFVB<VANDOB>OANDC(OB)=OTHENM$="NON HAI ""+W$+""
420 IFF(26)=1ANDRM=13ANDINT(RND(1)*3-1)<>3ANDVB<>21THENGOTO425
423 GOTO430
425 M$="PIPISTRELLI ALL'ATTACCO!":GOTO90
430 IFRM=44ANDINT(RND(1)*2+1)=1ANDF(24)<>1THENF(27)=1
440 IFF(0)=1THENLL=LL-1
450 IFLL<1THENF(0)=0
455 IFVB>14THENGOTO465
```

```
460 ONVBCOSUB500,570,640,640,640,640,640,980,980,1030,1070,1140
463 GOTO470
465 ONVB=14GOSUB1180,1220,1250,1300,1340,1380,1400,1430,1460,1490,1510,1590
```

```
470 IFLL=10THENM$="LA TUA CANDELA SI STA SPEGNENDO"
480 IFLL=1THENM$="LA TUA CANDELA SI E' SPENTA"
490 GOTO90
```

```
500 PRINT"PAROLE RICONOSCIUTE:"
510 FORI=1TOW
520 PRINTV$(I);", ";
530 NEXTI
540 M$="":PRINT
550 GOSUB1580
560 RETURN
```

```
570 PRINT"HAI CON TE:"
580 FORI=1TOW
590 IFC(I)=1THENPRINTO$(I);", ";
600 NEXTI
610 M$="":PRINT
620 GOSUB1580
630 RETURN
```

```
640 D=0
650 IFOB=OTHEND=VB-3
660 IFOB=19THEND=1
670 IFOB=20THEND=2
680 IFOB=21THEND=3
690 IFOB=22THEND=4
700 IFOB=23THEND=5
710 IFOB=24THEND=6
720 IFRM=20ANDD=5THEND=1
730 IFRM=20ANDD=6THEND=3
740 IFRM=22ANDD=6THEND=2
750 IFRM=22ANDD=5THEND=3
760 IFRM=36ANDD=6THEND=1
770 IFRM=36ANDD=5THEND=2
780 IFF(14)=1THENM$="CRASH! SEI CADUTO DALL'ALBERO":F(14)=0:RETURN
790 IFF(27)=1ANDRM=52THENM$="I FANTASMI TI IMPEDISCONO DI MUOVERTI":RETURN
800 IFRM=45ANDC(1)=1ANDF(34)=OTHENM$="UNA BARRIERA MAGICA AD OVEST":RETURN
810 IFRM=26ANDF(0)=0)AND(D=1ORD=4)THENM$="PER ANDARE DI LA' DEVI FARE LUCE":RETURN
```



Vedere alle pag. 23-24 per scoprire come funziona la sezione dei messaggi d'errore.



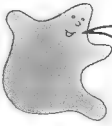
Usate qui l'equivalente istruzione di RND del vostro computer.



Cambiate questa linea su BBC. Vedere a pag. 37.



State estremamente attenti nell'inserire correttamente questa linea. Se la sbagliate, vi manderà all'aria il programma.



Le sezioni di salto alle subroutines ■ dei verbi sono spiegate alle pag. 24 e 25.



Potete vedere come funziona la subroutine ANDARE alle sub-pag. 26-27.

```

820 IFRM=54ANDC(15)<>1THENM$="SEI BLOCCATO!":RETURN
830 IFC(15)=1ANDNOT(RM=53ORRM=54ORRM=55ORRM=47)THENGOTO835
833 GOTO840
835 M$="NON PUOI TRASPORTARE UNA BARCA!":RETURN
840 IF(RM>26ANDRM<30)ANDF(0)-OTHERM$="TROPPO BUIO PER MUOVERSI":RETURN
850 F(35)=0:RL=LEN(R$(RM))
860 FORI=1TORL
870 U$=MID$(R$(RM),I,1)
880 IF(U$="N"ANDD=1ANDF(35)=0)THENRM=RM-8:F(35)=1
890 IF(U$="S"ANDD=2ANDF(35)=0)THENRM=RM+8:F(35)=1
900 IF(U$="O"ANDD=3ANDF(35)=0)THENRM=RM-1:F(35)=1
910 IF(U$="E"ANDD=4ANDF(35)=0)THENRM=RM+1:F(35)=1
920 NEXTI
930 M$="OK"
940 IFF(35)=0THENM$="NON PUOI ANDARE DA QUELLA PARTE!"
950 IFD<1THENM$="ANDARE DOVE?":

```



```

960 IFRM=41ANDF(23)=1THENR$(49)="SO":M$="LA PORTA SI CHIUDE DI COLPO!":F(23)=0
970 RETURN

```

```

980 IFOB>GTHENM$="NON POSSO PRENDERE "+W$:RETURN
985 IFL(OB)<>RTHENM$="NON CE N'E' QUI"
990 IFF(OB)<>0THENM$="QUALE "+W$+"?"
1000 IFC(OB)=1THENM$="CE L'HAI GIA'"
1010 IFOB>OANDL(OB)=RMANDF(OB)=0THENC(OB)=1:L(OB)=65:M$="HAI CON TE ":+W$
1020 RETURN

```

```

1030 IFRM=43AND(OB=28OROB=29)THENF(17)=0:M$="CASSETTO APERTO"
1040 IFRM=28ANDOB=25THENM$="E' CHIUSA A CHIAVE"
1050 IFRM=38ANDOB=32THENM$="E' SPETTRALE!":F(2)=0
1060 RETURN

```

```

1070 IFOB=30THENF(18)=0:M$="C'E' QUALCOSA QUI!"
1080 IFOB=31THENM$="E' DISGUSTOSO!"
1090 IF(OB=28OROB=29)THENM$="C'E' UN CASSETTO"
1100 IFOB=33OROB=5THENGOSUB1140
1110 IFRM=43ANDOB=35THENM$="C'E' QUALCOSA OLTRE..."
1120 IFOB=32THENGOSUB1030
1130 RETURN

```

```

1140 IFRM=42ANDOB=33THENM$="SONO OPERE DEL DEMONIO"
1150 IF(OB=3OROB=36)ANDC(3)=1ANDF(34)-OTHERM1155
1153 GOTO1160
1155 M$="USA QUESTA PAROLA CON CAUTELA:'XZANFAR'"
1160 IFC(5)=1ANDOB=5THENM$="LO SCRITTO E' IN UNA LINGUA SCONOSCIUTA"
1170 RETURN

```

Usate qui l'equivalente istruzione RND del vostro computer.



```

1180 M$="OK "+W$+"!"
1190 IFC(3)=1ANDOB=34THENM$="*AVVIENE UNA MAGIA*":IFRM<>45THENRM=INT(RND(1)*63+1)
1200 IFC(3)=1ANDOB=34ANDRM=45THENF(34)=1
1210 RETURN

```

```

1220 IFC(12)=1THENM$="HAI FATTO UN BUCO"
1230 IFC(12)=1ANDRM=30THENM$="DIVELTE LE SBARRE":D$(RM)="BUCO NEL MURO":R$(RM)="NSE"
1240 RETURN

```

```

1250 IFC(14)<>1ANDRM=7THENM$="NON E' QUESTO IL MOMENTO DI GIOCARE"
1260 IFOB=14ANDC(14)=1THENM$="L'HAI LANCIATA"
1270 IFOB=13ANDC(13)=1THENM$="WHOOSH!"

```

VERBO 10 E 11

VERBO 12

VERBO 13

VERBO 14

VERBO 15

VERBO 16

VERBO 17

1280 IFOB=13ANDC(13)=1ANDRM=43THENR\$(RM)="ON":GOTO1285
 1283 GOTO1290
 1285 D\$(RM)="STUDIO CON STANZA SEGRETA":M\$="HAI SFONDATO LA PARETE SOTTILE"
 1290 RETURN

1300 IFOB=14ANDC(14)=1THENM\$="NON E' ATTACCATA A NULLA!"
 1310 IFOB=1RANDC(14)<>1ANDRM=7ANDF(14)=0THEN1315
 1313 GOTO1320
 1315 M\$="VEDI UNA FORESTA SCURA ED UNA SCOGLIERA A SUD":F(14)=1:RETURN
 1320 IFOB=14ANDC(14)<>1ANDRM=7ANDF(14)=1THENM\$="STAI SCENDENDO!":F(14)=0
 1330 RETURN

1340 IFOB=17ANDC(17)=1ANDC(8)=0THENM\$="TI BRUCERA' LE MANI"
 1350 IFOB=17ANDC(17)=1ANDC(9)=0THENM\$="NIENTE CON CUI ACCENDERLA"
 1360 IFOB=17ANDC(17)=1ANDC(9)=1ANDC(8)=1THENM\$="SPANDE UNA LUCE TREMOLANTE":F(0)=1
 1370 RETURN

1380 IFF(0)=1THENF(0)=0:M\$="SPENTA"
 1390 RETURN

1400 IFOB=26ANDC(16)=1THENM\$="HISSSS"
 1410 IFOB=26ANDC(16)=1ANDF(26)=1THENF(26)=0:M\$="PFFT! LI HAI PRESI"
 1420 RETURN

1430 IFOB=10ANDC(10)=1ANDC(11)=1THENM\$="ACCESA":F(24)=1
 1440 IFF(27)=1ANDF(24)=1THENM\$="WHIZZ-HAI RISUCCHIATO I FANTASMI":F(27)=0
 1450 RETURN

1460 IFRM=43AND(OB=27OROB=28)THENGO SUB1030
 1470 IFRM=28ANDOB=25ANDF(25)=0ANDC(18)=1THENF(25)=1:R\$(RM)="SEO":GOTO1475
 1473 GOTO1480
 1475 D\$(RM)="ENORME PORTA APERTA":M\$="LA CHIAVE GIRA!"
 1480 RETURN

1490 IFC(OB)=1THENC(OB)=0:L(OB)=RM:M\$="FATTO"
 1500 RETURN

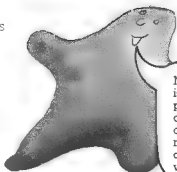
1510 S=0
 1520 FORI=1TOG
 1530 IFC(I)=1THENS=S+1
 1540 NEXTI
 1550 IFS=17ANDC(15)<>1ANDRM<>57THENPRINT"HAI PRESO TUTTO":GOTO1555
 1553 GOTO1560
 1555 PRINT"RITORNA AL SENTIERO DI INGRESSO PER IL PUNTEGGIO FINALE"
 1560 IFS=17ANDRM=57THENPRINT"PUNTEGGIO RADDOPPIATO PER ESSERE ARRIVATO FIN QUI":S=S*2
 1570 PRINT"IL TUO PUNTEGGIO= ";S:IFS>18THENPRINT"COMPLIMENTI! HAI TERMINATO IL GIOCO":END
 1580 INPUT"PREMI RETURN PER CONTINUARE":Q\$
 1590 RETURN

1600 DIMR\$(63),D\$(63),O\$(W),V\$(V)
 1610 DIMC(W),L(G),F(W)
 1620 DATA46,38,35,50,13,18,28,42,10,25,26,4,2,7,47,60,43,32
 1630 FORI=1TOG
 1640 READL(I)
 1650 NEXTI
 1660 DATAAIUTO,POSSESSO?,ANDARE,N,S,O,E,A,B,PIGLIARE,PRENDERE,APRIRE,ESAMINARE
 1665 DATALEGGERE,DIRE,SCAVARE,BRANDIRE,SALIRE,ACCENDERE,SPEGNERE,SPRUZZARE
 1670 DATAUSARE,DISSERRARE,LASCIARE,PUNTEGGIO
 1680 FORI=1TOV
 1690 READV\$(I)
 1700 NEXTI



Se avete un BBC, qui potete aver bisogno di una virgola al posto di punto e virgola.

1730 DATANS,NS,SE,OE,NOAB,SE,OSAB,NS
 1740 DATAN,NS,NSE,OE,OE,NSO,NS,NS
 1750 DATAS,NSE,NSO,S,NSAB,N,N,NS
 1760 DATANE,NO,NE,O,NSE,OE,O,NS
 1770 DATASE,NSO,E,OE,NO,S,SO,NO
 1780 DATANE,NOE,OE,OE,OE,NOE,NOE,O
 1790 FORI-OTO63
 1800 READR\$(1)
 1810 NEXTI



Notare che i termini in un istruzione di dati sono separati da una virgola. Se cambiate i dati, assicuratevi di non includere come nuovi termini delle virgole, o confonderete le idee al vostro computer.

1820 DATA ANGOLO BUIO,GIARDINO INCOLTO,ACCANTO AD UNA GRANDE CATASTA DI LEGNA
 1825 DATA CORTILE CON DEI RIFIUTI,MACCHIA DI ERBACCE,FORESTA,IN MEZZO ALLA FORESTA
 1830 DATA ALBERO FULMINATO,ANGOLO DELLA CASA,INGRESSO DELLA CUCINA
 1835 DATA CUCINA CON FORNELLO SUDICIO,PORTA DEL RETROCUCINA
 1840 DATA STANZA CON UNO STRATO DI POLVERE,STANZA DELLA TORRE SUL RETRO
 1845 DATA SPIAZZO ACCANTO ALLA CASA,SENTIERO,LATO DELLA CASA,IN FONDO AL SALONE
 1850 DATA ALCOVA BUIA,PICCOLA STANZA BUIA,AI PIEDI DELLA SCALA A CHIOCCIOLA
 1855 DATA AMPIO CORRIDOIO,SCALINI SCIVOLOSI,CIMA DELLA SCOGLIERA
 1860 DATA ACCANTO ALLE MURA PERICOLANTI,CORRIDOIO TETRO,CERCHIO DI LUCE
 1865 DATA GRANDE SALONE CON SOFFITTO A VOLTA,SALONE CON GROSSA PORTA DI LEGNO
 1870 DATA STANZA DEI TROFEI,CANTINA CON FINESTRA SBARRATA,SENTIERO SULLA SCOGLIERA
 1875 DATA ARMADIO CON CAPPOTTO APPESO,SALONE DI INGRESSO,SALOTTO,STANZA SEGRETA
 1880 DATA RIPIDE SCALE DI MARMO,SALA DA PRANZO,IN FONDO ALLA CANTINA CON BARA
 1885 DATA SENTIERO SULLA SCOGLIERA,SGABUZZINO,ATRIO D'INGRESSO
 1890 DATA BIBLIOTECA CON LIBRI DEL MALE,STUDIO CON SCRIVANIA E BUCO NEL MURO
 1895 DATA STRANA STANZA PIENA DI RAGNATELE,CAMERA MOLTO FREDDA,STANZA SPETTRALE
 1900 DATA SENTIERO SULLA SCOGLIERA ACCANTO ALLA PALUDE
 1905 DATA VERANDA COSPASA DI PIETRISCO,PORTICO D'INGRESSO,TORRE SUL DAVANTI
 1910 DATA CORRIDOIO IN PENDENZA,PASSAGGIO SUPERIORE,PALUDE ACCANTO ALLE MURA
 1915 DATA PALUDE,SENTIERO MELMOSO,ACCANTO ALLA RINGHIERA CONTORTA
 1920 DATA SENTIERO PER IL CANCELLO DI FERRO,ACCANTO ALLA RINGHIERA
 1925 DATA SOTTO LA TORRE SUL DAVANTI,DETRITTI DALLA FACCIATA SGRETOLANTE
 1930 DATA GROSSI CALCINACCI CADUTI,ARCO DI PIETRA IN ROVINA,SCOGLIERA FRANANTE
 1980 FORI-OTO63
 1990 READR\$(1)
 2000 NEXTI
 2010 DATA IL QUADRO,L'ANELLO,INCANTESIMI MAGICI,IL CALICE,IL PAPANO,LE ANTE
 2015 DATA LA STATUA,IL CANDELIERE,I FIAMMIFERI,L'ASPIRAPOLVERE,LE PILE,LA LAMPA
 2020 DATA L'ASCIA,LA CORDA,LA BARCA,L'AEROSOL,LA CANDELA,LA CHIAVE,A NORD,IL SUB
 2025 DATA A OVEST,A EST,IN ALTO,IN BASSO,LA PORTA,I PIPISTRELLI,I FANTASMI
 2030 DATA IL CASSETTO,LA SCRIVANIA,IL CAPPOTTO,I RIFIUTI,LA BARA,I LIBRI
 2035 DATA IL MURO,GLI INCANTESIMI
 2060 FORI-OTOW
 2070 READR\$(1)
 2080 NEXTI



Assicuratevi di inserire i dati nell'ordine giusto, o accadranno strane cose quando cercherete di giocare.

2090 F(18)-1:F(17)-1:F(2)-1:F(26)-1:F(28)-1:F(23)-1:LL=60:RM=57:M\$="OK"
 2100 RETURN

Modifiche per BBC

Se avete un BBC, usate queste linee al posto della linea 460 nel listato principale.

460 ON VB GOSUB 500,570,640,640,640,640,640,640,980,980,1030
 ,1070,1140,1180,1220,1250,1300,1340,1380,1400,1430,1460,1490,1510,1590

Modifiche per lo Spectrum (Timex 2000)

I computers Sinclair (Timex) usano una versione di BASIC che differisce molto dal BASIC di altri computer molto diffusi, per cui dovrete effettuare un bel po' di modifiche al listato, per farvelo girare. Queste modifiche rendono il programma un po' troppo lungo per l'inserimento in uno Spectrum 16K. Peraltro, potreste cercare di adattare il programma perché tale inserimento sia possibile, eliminando alcuni verbi, ad esempio, ed abbreviando i messaggi.

Utilizzate le linee listate sotto per sostituire quelle corrispondenti del

```
60 LEFT$=""
120 PRINT D$(RM+1)
140 FOR I=1 TO LEN(R$(RM+1))
150 PRINT R$(RM+1) (I TO I);", ";
240 LET X$="":LET W$="":LET VB=0:LET OB=0
250 FOR I=1 TO LEN(Q$)-1
260 IF Q$(I TO I)="" AND X$="" THEN LET X$=Q$(I TO I)
270 IF Q$(I+1 TO I+1)<>" " AND X$="" THEN LET W$=Q$(I+1 TO I):LET I=LEN(Q$)-1
```

programma principale e modificate tale programma come segue:

1. Lo Spectrum deve avere inserito il comando LET ogni volta che assegnate un valore ad una variabile, ad esempio LET V=25. Attenzione questo influenza molte linee, incluse quelle contenenti IF ... THEN.
2. Tutti i dati di tipo stringa di caratteri alle linee 1660, 1665, 1710-1780, 1820-1965 e 2010-2018, sono tra virgolette, così: 1820 DATA "ANGOLO BUIO", "GIARDINO INCOLTO", etc.
3. Alle linee 1790 e 1980, cambiate il ciclo in modo che inizi con FOR I=1 TO 64 (lo Spectrum non permette l'uso della casella zero in una matrice).

Se inserite i dati in lettere maiuscole, dovrete giocare usando le maiuscole — il computer non riconosce che "ANDARE A OVEST" e "andare a ovest" sono la stessa cosa. È meglio tenere inserita la funzione di CAPS LOCK (blocco delle maiuscole).



```
290 IF W$="" THEN LET X$=Q$
295 IF LEN(X$)>LEN(V$(1)) OR X$="" THEN GOTO 325
296 LET X$=X$+F$( TO (LEN(V$(1))-LEN(X$)))
310 IF X$=V$(1) THEN LET VB=1
325 IF W$="" OR LEN(W$)>LEN(O$(1)) THEN GOTO 360
326 LET W$=W$+F$( TO (LEN(O$(1))-LEN(W$)))
405 IF OB=0 THEN GOTO 420
420 IF F(26)=1 AND RM=13 AND INT(RND*3)+1<>3 AND VB<21 THEN LET M$="FANTASMI ALL'ATTACCO"
430 IF RM=44 AND INT(RND*2)=1 AND F(24)<1 THEN LET F(27)=1
440 IF F(20)=1 THEN LET LL=LL-1
450 IF LL<1 THEN LET F(20)=0
460 GOSUB 500*(VB=1)+570*(VB=2 AND VB<10)+980*(VB=10 OR VB=11)+1030*(VB=12)+1070*(VB=13)+1140*(VB=14)+1180*(VB=15)+1220*(VB=16)+1250*(VB=17)+1300*(VB=18)+1340*(VB=19)+1380*(VB=20)+1400*(VB=21)+1430*(VB=22)+1460*(VB=23)+1490*(VB=24)+1510*(VB=25)+1590*(VB=26)
600 IF (RM=26 AND F(20)=0) AND (D=1 OR D=4) THEN LET M$="PER ANDARE DI LA' DEVI FARE LUCE":RETURN
840 IF (RM>26 AND RM<30) AND F(20)=0 THEN LET M$="TROPPO BUIO PER MUOVERSI":RETURN
850 LET F(35)=0:LET RL=LEN(R$(RM+1))
870 LET U$=R$(RM+1) (I TO I)
960 IF RM=41 AND F(23)=1 THEN LET R$(50)="SO":LET M$="LA PORTA SI CHIUDE DI COLPO!":LET F(23)=0
980 IF OB>G OR OB=0 THEN LET M$="NON POSSO PRENDERE "+W$:RETURN
1190 IF C(3)=1 AND OB=34 THEN LET M$="ACCADDE UNA MAGIA":IF RM<45 THEN LET RM=INT(RND*64)
1230 IF C(12)=1 AND RM=30 THEN LET M$="DIVELTE LE SBARRE":LET D$(RM+1)="BUCO NEL MURO":LET R$(RM+1)="NSE"
1280 IF OB=13 AND C(13)=1 AND RM=43 THEN LET R$(RM+1)="ON":LET D$(RM+1)="STUDIO CON STANZA SEGRETA":LET M$="HAI SFONDATO LA PARETE SOTTILE"
1350 IF OB=17 AND C(17)=1 AND C(9)=1 AND C(8)=1 THEN LET M$="SPANDE UNA LUCE TREMOLANTE":LET F(20)=1
1380 IF F(20)=1 THEN LET F(20)=0:LET M$="SPENTA"
1470 IF RM=28 AND OB=25 AND F(25)=0 AND C(18)=1 THEN LET F(25)=1:LET R$(RM+1)="SEO":LET D$(RM+1)="ENORME PORTA APERTA":LET M$="LA CHIAVE GIRAI"
1600 DIM R$(64,4):DIM D$(64,4):DIM O$(W,13):DIM V$(V,9)
```

Versione per ZX81 (Timex 1000)



















Il listato del programma alle sei pagine successive è una versione speciale della "Casa Stregata" per lo ZX81. Essa corrisponde il più possibile alla struttura del programma principale, così potrete seguire le spiegazioni del programma date nel libro. Le differenze principali

sono: lo ZX81 potrà accettare solo una istruzione per linea, e non ha i comandi READ ... DATA. Il programma è stato riscritto per tener conto di queste ed altre differenze nel BASIC, che vedrete annotate accanto al listato. Questi cambiamenti portano via molto spazio di memoria. Per farci stare dentro il programma del gioco il numero di locazioni è stato ridotto da 64 a 36, ed altri dati sono stati cambiati leggermente, per adeguarsi a questa riduzione. La versione per lo ZX81 dello schema principale è mostrata nel seguito. Potete vedere come i dati vengano inseriti nella memoria del computer nel seguente paragrafo.



Notare che le locazioni vengono numerate a partire da 1, in quanto lo ZX81 non permette l'uso della casella zero nelle matrici.

Schema principale per ZX81

1 INGRESSO DELLA CUCINA	2 CUCINA 	3 RETROCUCINA 	4 STANZA POLVEROSA	5 STANZA DELLA TORRE SUL RETRO 	6 ARMADIO CON CAPPOTTO 
7 IN FONDO AL SALONE	8 ALCOVA BUIA 	9 STANZINO CON RIFIUTI	10 SCALA A CHIOCCIOLA	11 AMPIO CORRIDOIO 	12 SCALINI SCIVOLOSI
13 SALONE VERSO IL RETRO 	14 CERCHIO DI LUCE 	15 SALA CON SOFFITTO A VOLTA	16 SALA CON PORTA SBARRATA 	17 STANZA DEI TROFEI	18 CANTINA 
19 SALONE D'INGRESSO	20 SALOTTO	21 STANZA SEGRETA 	22 RIPIDE SCALE DI MARMO	23 SALA DA PRANZO	24 SOTTERRANEO CON BARA 
25 ATRIO D'INGRESSO	26 BIBLIOTECA 	27 STUDIO 	28 STANZA PIENA DI RAGNATELE	29 CAMERA FREDDA	30 STANZA SPETTRALE 
31 PORTICO D'INGRESSO	32 TORRE SUL DAVANTI 	33 CORRIDOIO IN PENDINGENZA	34 PASSAGGIO SUPERIORE	35 DEPOSITO BARCA 	36 SENTIERO MELMOSO 

Come usare il programma

Se esaminate il listato vi accorgete che i dati per il gioco non sono inclusi nel programma. Il programma funziona chiedendovi di inserire i dati da tastiera, e poi salvando l'intero programma, con i dati, su nastro. Questo va fatto solo una volta — la volta successiva dovete solo caricare il tutto dal nastro.

Istruzioni per usare il programma:

1. Inserite il programma da tastiera (con molta attenzione).
2. Inserite: RUN 2440.
3. Battete i dati nel seguente ordine
4. Ora salvate il programma su nastro con SAVE. Quest'operazione salverà anche tutti i dati.
5. Per iniziare il gioco, battete GOTO 10. **NON BATTETE RUN**: questo distruggerebbe il contenuto delle variabili.
6. Ora inserite le posizioni iniziali degli oggetti. Quando avete inserito l'ultima, il programma vi darà la locazione iniziale.
7. Per ripetere il gioco, ripetere i comandi ai punti 5 e 6.
8. Caricando il programma da nastro, seguite queste istruzioni dal punto 5.

```

10 GOSUB 2200
20 CLS
30 PRINT "L'AVVENTURA DELLA CASA STREGATA"
40 PRINT "-----"
50 PRINT "LA TUA LOCAZIONE:"
60 PRINT D$(RM)
70 PRINT "USCITE:"
80 FOR I=1 TO LEN(R$(RM))
90 PRINT R$(RM)(I TO I);", ";
100 NEXT I
110 PRINT
120 FOR I=1 TO G
130 IF L(I)-RM AND F(I)=0 THEN PRINT "QUI PUOI TROVARE ";O$(I)
140 NEXT I
150 PRINT "-----"
160 PRINT M$
170 LET M$="COSA ?"

```

```

180 PRINT "COSA VUOI FARE ADESSO?"
190 INPUT Q$
200 LET X$=""
210 LET W$=""
220 LET VB=0
230 LET OB=0
240 FOR I=1 TO LEN(Q$)-1
250 IF Q$(I TO I)=" " AND X$="" THEN LET X$=Q$(TO I-1)
260 IF Q$(I+1 TO I+1)<>" " AND X$<>" " THEN LET W$=Q$(I+1 TO I)
270 IF W$<>" " THEN LET I=LEN(Q$)-1
280 NEXT I
290 IF W$="" THEN LET X$=Q$
300 IF LEN(X$)>LEN(V$(1)) OR X$="" THEN GOTO 420
310 LET F=LEN(V$(1))-LEN(X$)
320 LET X$=X$+F$(TO F)
330 FOR I=1 TO V
340 IF X$=V$(I) THEN LET VB=I
350 NEXT I
360 IF W$="" OR LEN(W$)>LEN(O$(1)) THEN GOTO 430

```

Lo ZX81 ha bisogno del comando LET per assegnare un valore ad una variabile.

Questa sezione appare un po' diversa che non sul programma principale, poiché lo ZX81 non usa le funzioni MID\$, LEFT\$ e RIGHT\$ per sezionare le stringhe.

```

370 LET F=LEN(O$(1))-LEN(W$)
380 LET W$=W$+F$(TO F)
390 FOR I=1 TO W
400 IF W$=O$(I) THEN LET OB=I
410 NEXT I

```

```

420 IF W$>" AND OB=0 THEN LET M$="CHE SCIOCCHEZZA"
430 IF VB=0 THEN LET VB=V+1
440 IF W$="" THEN LET M$="MI SERVONO DUE PAROLE"
450 IF VB>V AND OB>0 THEN LET M$="NON PUOI "+Q$
460 IF VB>V AND OB=0 THEN LET M$="QUESTO NON HA SENSO"
470 IF OB=0 OR OB>G THEN GOTO 490
480 IF VB<V AND OB>0 AND C(OB)=0 THEN LET M$="NON HAI "+W$
490 IF F(26)=0 OR RM<>5 OR INT(RND*3)-2 OR VB=21 THEN GOTO 520
500 LET M$="PIPISTRELLI ALL'ATTACCO"
510 GOTO 20
520 IF RM=28 AND INT(RND*2)-1 AND F(24)=0 THEN LET F(27)=1
530 IF F(20)=1 THEN LET LL=LL-1
540 IF LL<1 THEN LET F(20)=0

```

Notare che alcune delle linee di programma sono più lunghe delle linee di stampa sulla pagina. State attenti a non premere NEWLINE prima della fine della linea di programma.



```

550 GOSUB 590*(VB=1)+660*(VB=2)+730*(VB>2 AND VB<10)+1160*(VB=10
OR VB=11)+1270*(VB=12)+1350*(VB=13)+1440*(VB=14)+1480*(VB=15)+1540*
(VB=16)+1560*(VB=17)+1640*(VB=18)+1700*(VB=19)+1760*(VB=20)+1800*(V
B=21)+1850*(VB=22)+1920*(VB=23)+1990*(VB=24)+2040*(VB=25)+2190*(VB=26)

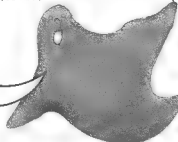
```

```

560 IF LL=10 THEN LET M$="LA TUA CANDELA SI STA SPENGENDO"
570 IF LL=1 THEN LET M$="LA TUA CANDELA SI E' SPENTA"
580 GOTO 20

```

Questa linea sostituisce la linea di ON...GOSUB, che lo ZX81 non può eseguire. Funziona come un'operazione di calcolo, utilizzando il valore di VB. Il computer esamina il contenuto di ciascuna parentesi con "VB=", e pone un valore 1 se l'equazione logica tra parentesi è "vera", 0 se è "falsa". Provate ora ad eseguire il calcolo assegnando a VB un valore reale.



```

590 PRINT "PAROLE RICONOSCIUTE"
600 FOR I=1 TO V
610 PRINT V$(I);", ";
620 NEXT I
630 LET M$=""
640 GOSUB 2160
650 RETURN

```

```

660 PRINT "HAI CON TE:"
670 FOR I=1 TO G
680 IF C(I)=1 THEN PRINT O$(I);", ";
690 NEXT I
700 LET M$=""
710 GOSUB 2160
720 RETURN

```

```

730 LET D=0
740 IF OB=0 THEN LET D=VB-3
750 IF OB>18 AND OB<25 THEN LET D=OB-18
760 IF RM=10 AND D=5 THEN LET D=1
770 IF RM=10 AND D=6 THEN LET D=3
780 IF RM=12 AND D=6 THEN LET D=2
790 IF RM=12 AND D=5 THEN LET D=3
800 IF RM=22 AND D=6 THEN LET D=1
810 IF RM=22 AND D=5 THEN LET D=2
820 IF RM<>32 OR D<>3 THEN GOTO 850
830 LET M$="E' UN BEL SALTO"

```

Questa sostituisce la sezione dell'albero, nel programma principale. Controllate lo schema se volete vedere qual'è la locazione 32.

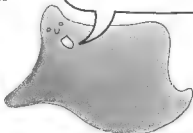


```

840 RETURN
850 IF F(27)=0 OR RM<34 THEN GOTO 880
860 LET M$="I FANTASMI TI IMPEDISCONO DI MUOVERTI"
870 RETURN
880 IF RM<29 OR C(1)=0 OR F(34)=1 THEN GOTO 910
890 LET M$="UNA BARRIERA MAGICA AD OVEST"
900 RETURN
910 IF RM<14 OR RM>17 OR F(20)=1 THEN GOTO 950
920 IF RM=14 AND D<>1 AND D<>4 THEN GOTO 950
930 LET M$="TROPPO BUIO PER MUOVERSI"
940 RETURN
950 IF C(15)=0 OR RM<>36 THEN GOTO 980
960 LET M$="LA BARCA E' TROPPO PESANTE"
970 RETURN
980 LET RL=LEN(R$(RM))
990 LET OM=RM
1000 FOR I=1 TO RL
1010 LET U$=R$(RM)(I TO I)
1020 IF U$="N" AND D=1 THEN LET OM=OM-6
1030 IF U$="S" AND D=2 THEN LET OM=OM+6
1040 IF U$="O" AND D=3 THEN LET OM=OM-1
1050 IF U$="E" AND D=4 THEN LET OM=OM+1
1060 NEXT I
1070 LET M$="OK"
1080 IF RM=-OM THEN LET M$="NON PUOI ANDARE PER DI LA'"
1090 LET RM=OM
1100 IF D<1 THEN LET M$="ANDARE DOVE ?"
1110 IF RM<>25 OR F(23)=0 THEN GOTO 1150
1120 LET R$(31)=" "
1130 LET M$="LA PORTA SI CHIUDE DI COLPO DIETRO DI TE"
1140 LET F(23)=0
1150 RETURN

```

Controllate ciascuna linea prima di premere NEWLINE. È molto più facile cercare di individuare gli errori man mano che si inseriscono le linee, che non il doverli poi cercare esaminando l'intero listato.



Vedere a pag. 24-25 per altre notizie sul funzionamento delle subroutines.



```

1160 IF OB>0 AND OB<G THEN GOTO 1190
1170 LET M$="NON PUOI PRENDERE "+W$
1180 RETURN
1190 IF L(OB)<>RM THEN LET M$="NON E' QUI"
1200 IF F(OB)=1 THEN LET M$="QUALE "+W$+" ?"
1210 IF C(OB)=1 THEN LET M$="CE L'HAI GIA'"
1220 IF L(OB)<>RM OR F(OB)=1 THEN GOTO 1260
1230 LET C(OB)=1
1240 LET M$="HAI CON TE "+W$
1250 LET L(OB)=37
1260 RETURN

```

Immaginate che il vostro gioco d'avventura debba essere venduto in negozio, ed illustrate con disegni e descrizioni il cartoncino per la cassetta del programma.



```

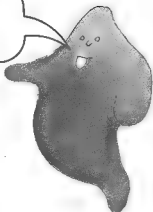
1270 IF RM<>27 OR (OB<>28 AND OB<>29) THEN GOTO 1300
1280 LET M$="CASSETTO APERTO"
1290 LET F(17)=0
1300 IF RM=16 AND OB=25 THEN LET M$="E' CHIUSO A CHIAVE"
1310 IF RM<>24 OR OB<>32 THEN GOTO 1340
1320 LET M$="SPETTRALE!"

```

1330 LET F(2)=0
1340 RETURN

1350 IF OB<>30 THEN GOTO 1380
1360 LET M\$="C'E' QUALCOSA QUI"
1370 LET F(18)=0
1380 IF OB=26 OR OB=29 THEN LET M\$="C'E' UN CASSETTO"
1390 IF OB=33 OR OB=5 THEN GOSUB 1440
1400 IF RM=27 AND OB=35 THEN LET M\$="C'E' QUALCOSA OLTRE"
1410 IF OB=32 THEN GOSUB 1270
1420 IF RM=9 AND OB=31 THEN LET M\$="E' DISGUSTOSO"
1430 RETURN

Confrontate tra di loro le due versioni della "Casa Stregata" per vedere dove differiscono.



1440 IF RMOR26 AND OB=33 THEN LET M\$="SONO OPERE DEL DEMONIO"
1450 IF (OB=3 OR OB=36) AND C(3)=1 AND F(34)=0 THEN LET M\$="USA QUESTA PAROLA CON CAUTELA - XZANFAR"
1460 IF C(5)=1 AND OB=5 THEN LET M\$="UNA LINGUA SCONOSCIUTA"
1470 RETURN

1480 LET M\$="OK "+W\$
1490 IF C(3)=0 OR OB<>34 THEN GOTO 1520
1500 LET M\$="AVVIENE UNA MAGIA"
1510 IF RM<>29 THEN LET RM=INT(RND*36)+1
1520 IF C(3)=1 AND OB=34 AND RM=29 THEN LET F(34)=1
1530 RETURN

Potete anche aggiungere altri verbi senza aggiungere subroutines. Potrete aggiungere ad esempio GUARDARE, e far usare a questo la routine di ESAMINARE.

1540 IF C(12)OR1 THEN LET M\$="HAI FATTO UN BUCO"
1550 RETURN

1560 IF C(14)<>1 AND RM=11 THEN LET M\$="QUESTO NON E' IL MOMENTO DI GIOCARE"
1570 IF OB=14 AND C(14)=1 THEN LET M\$="L'HAI LANCIATA"
1580 IF OB=13 AND C(13)=1 THEN LET M\$="WHOOOSH"
1590 IF OB<>13 OR C(13)=0 OR RM<>27 THEN GOTO 1630
1600 LET R\$(RM)="ON"
1610 LET D\$(M)="STUDIO CON STANZA SEGRETA"
1620 LET M\$="HAI APERTO UN PASSAGGIO"
1630 RETURN



1640 IF RM<>32 OR C(14)<>1 OR OB<>14 THEN GOTO 1680
1650 LET M\$="STAI SCENDENDO"
1660 LET RM=RM-1
1670 GOTO 1690
1680 LET M\$="DOVE ?"
1690 RETURN

1700 IF OB=17 AND C(17)=1 AND C(8)=0 THEN LET M\$="TI BRUCERA' LE MANI"
1710 IF OB=17 AND C(17)=1 AND C(9)=0 THEN LET M\$="CON CHE COSA?"
1720 IF OB<>17 OR C(17)=0 OR C(9)=0 OR C(8)=0 THEN GOTO 1750
1730 LET M\$="SPANDE UNA LUCE TREMOLANTE"
1740 LET F(20)=1
1750 RETURN

1760 IF F(20)=0 THEN GOTO 1790
1770 LET M\$="SPENTA"
1780 LET F(20)=0
1790 RETURN

Forse potreste trovare un verbo migliore di "SPEGNERE".



1800 IF OB=26 AND C(16)=1 THEN LET M\$="HISSSS"

VERBO 13
VERBO 14
VERBO 15
16
VERBO 17
VERBO 18
VERBO 19
VERBO 20

```

1810 IF OB<>26 OR C(16)<>1 OR F(26)=0 THEN GOTO 1840
1820 LET M$="PFFT-LI HAI PRESI"
1830 LET F(26)=0
1840 RETURN

```

Le linee da 2300 a 2350 inizializzano gli indicatori per gli oggetti invisibili, così on avrete bisogno di inserire separatamente i dati di questi ultimi.

```

1850 IF OB<>10 OR C(10)=0 OR C(11)=0 THEN GOTO 1910
1860 LET F(24)=1
1870 LET M$="ACCESA"
1880 IF F(27)=0 OR F(24)=0 THEN GOTO 1910
1900 LET F(27)=0
1910 RETURN

```

```

1920 IF RM=27 AND (OB=27 OR OB =28) THEN GOSUB1270
1930 IF RM<>16 OR OB<>25 OR F(25)=1 OR C(18)=0 THEN GOTO1980
1940 LET F(25)=1
1950 LET M$="LA CHIAVE GIRA - CLUNK"
1960 LET R$(RM)="SEO"
1970 LET D$(RM)="GRANDE PORTA APERTA"
1980 RETURN

```

```

1990 IF C(OB)=0 THEN GOTO 2030
2000 LET C(OB)=0
2010 LET M$="FATTO"
2020 LET L(OB)=RM
2030 RETURN

```

Quando dimensionate una matrice di stringhe sullo ZX81, dovete render nota al computer la lunghezza del termine più lungo che vi immagazzinerete. Il computer allora riserverà 36 (o quante che siano) caselle di tale lunghezza. Questo porterà a spreco di memoria nel caso abbiate un termine molto più lungo di tutti gli altri.

```

2040 LET S=0
2050 FOR I=1 TO G
2060 IF C(I)=1 THEN LET S=S+1
2070 NEXT I
2080 IF S=17 AND C(15)=0 AND RM<>31 THEN PRINT "HAI PRESO TUT
TO. RITORNA AL PORTICO PER IL PUNTEGGIO FINALE"
2090 IF S<>17 OR RM<>31 THEN GOTO 2120
2100 PRINT " PUNTEGGIO RADDOPPIATO"
2110 LET S=S*2
2120 PRINT "IL TUO PUNTEGGIO= ";S
82 IF S<18 THEN GOTO 2160
2140 PRINT "COMPLIMENTI - HAI FINITO IL GIOCO"
2150 STOP
2160 PRINT "PREMI NEWLINE PER CONTINUARE"
2170 INPUT Q$
2180 LET M$="OK"
2190 RETURN

```

```

2200 FOR I=1 TO W
2210 LET F(I)=0
2220 LET C(I)=0
2230 NEXT I
2240 LET R$(31)="N"
2250 LET R$(27)="O"
2260 LET R$(16)="OE"
2270 LET D$(27)="STUDIO, SCRANIA E MURO"
2280 LET D$(16)="SALA CON GRANDE PORTA DI LEGNO"

```

Le linee da 2240 a 2380 riportano alle condizioni iniziali la variabili che sono state cambiate durante il gioco, così da permettervi di giocare di nuovo.


```

2290 LET M$="OK"
2300 LET F(18)=1
2310 LET F(17)=1
2320 LET F(27)=1
2330 LET F(2)=1
2340 LET F(26)=1
2350 LET F(23)=1
2360 LET LL=60
2370 LET RM=31
2380 LET F$=" ":REM 13 SPAZI
2390 FOR I=1 TO G
2400 PRINT I
2410 INPUT L(I)
2420 NEXT I
2430 RETURN
    
```

```

2440 DIM R$(36,4)
2450 DIM D$(36,30)
2460 LET V=25
2470 DIM V$(V,9)
2480 LET W=36
2490 DIM O$(W,13)
2500 DIM C(W)
2510 DIM F(W)
2520 LET G=18
2530 DIM L(G)
2540 PRINT "DESCRIZIONI"
2550 FOR I=1 TO 36
2560 PRINT I
2570 INPUT D$(I)
2580 NEXT I
2590 STOP
2600 PRINT "PERCORSI"
2610 FOR I=1 TO 36
2620 PRINT I
2630 INPUT R$(I)
2640 NEXT I
2650 STOP
2660 PRINT "OGGETTI"
2670 FOR I=1 TO W
2680 PRINT I
2690 INPUT O$(I)
2700 NEXT I
2710 STOP
2720 PRINT "VERBI"
2730 FOR I=1 TO V
2740 PRINT I
2750 INPUT V$(I)
2760 NEXT I
2770 STOP
    
```

Descrizioni

INGRESSO DELLA CUCINA, CUCINA CON FORNELLO SUDICIO, RETROCUCINA, STANZA POLVEROSA, STANZA DELLA TORRE SUL RETRO, ARMADIO CON CAPPOTTO, IN FONDO AL SALONE, ALCOVA BUIA, STANZINO CONRIFIUTI, SCALA A CHIOCCIOLA, AMPIO CORRIDOIO, SCALINI SCIVOLOSI, SALONE VERSO IL RETRO, CERCHIO DI LUCE, SALA CON SOFFITTO A VOLTA, SALA CON ENORME PORTA DI LEGNO, STANZA DEI TROFEI, CANTINA, SALONE D'INGRESSO, SALOTTO, STANZA SEGRETA, RIPIDE SCALE DI MARMO, SALA DA PRANZO, SOTTERRANEO CON BARA, ATRIO D'INGRESSO, BIBLIOTECA CON LIBRI DEL MALE, STUDIO CON SCRIVANIA E UN BUCO NEL URO, STANZA PIENA DI RAGNATELE, CAMERA MOLTO FREDDA, STANZA SPETTRALE, PORTICO D'INGRESSO, CIMA DELLA TORRE SUL DAVANTI, CORRIDORO IN PENDENZA, PASSAGGIO SUPERIORE, DEPOSITO BARCA, SENTIERO MELMOSO.

Percorsi

SE, OE, O, SE, OE, O, NS, SE, OE, NOAB, SE, SOAB, NS, NSE, OE, OE, NSO, NS, NSE, NSO, S, NSAB, N, N, N, NE, O, NSE, OE, SO, N, OE, OE, NO, E, NO.

Le virgole indicano dove dovete premere NEWLINE tra dati diversi. NON INSERITE LE VIRGOLE STESSE.

Oggetti

IL QUADRO, L'ANELLO, INCANTESIMI MAGICI, IL CALICE, IL PAPIRO, LE MONETE, LA STATUA, IL CANDELIERE, I FIAMMIFERI, L'ASPIRAPOLVERE, LE PILE, LA VANGA, L'ASCIÀ, LA CORDA, LA BARA, L'AEROSOL, LA CANDELA, LA CHIAVE, A NORD, A SUD, A OVEST, A EST, IN ALTO, IN BASSO, LA PORTA, I PIPISTRELLI, I FANTASMI, IL CASSETTO, LA SCRIVANIA, IL CAPPOTTO, I RIFIUTI, LA BARA, I LIBRI, XZANFAR, IL MURO, GLI INCANTESIMI.

Verbi

AUTO, POSSESSO, ANDARE, N, S, O, E, A, B, PIGLIARE, PRENDERE, APRIRE, ESAMINARE, LEGGERE, DIRE, SCAVARE, BRANDIRE, SCENDERE, ACCENDERE, SPEGNERE, SPRUZZARE, USARE, DISSERRARE, LASCIARE, PUNTEGGIO, LASCIARE.

Locazioni iniziali per gli oggetti

(Dovete inserire questa serie di dati ogni volta che fate girare il programma).
30, 24, 21, 32, 5, 8, 16, 26, 2, 13, 14, 36, 18, 11, 35, 3, 27, 6.

Ulteriori suggerimenti ed indicazioni

1. Usare variabili intere

Su certi computers potete mettere il simbolo % dopo il nome di una variabile numerica, per indicare che volete attribuire ad essa solo valori interi (ovvero numeri senza parte decimale a destra del punto). Così la variabile V diventerebbe V%, ■ così via. Questo potete farlo sui computers BBC, VIC 20, C 64. È utile farlo poiché fa risparmiare spazio in memoria e aumenta la velocità di esecuzione, anche del 50%. La velocità è particolarmente evidente quando il computer esegue dei cicli piuttosto lunghi.

2. Presentazione sul video

Se avete ancora abbastanza spazio in memoria, potreste migliorare l'immagine grafica con cui il gioco si presenta. Ad esempio, potreste aggiungere una routine grafica per il titolo d'apertura, e far lampeggiare le parole del testo in certi punti del gioco, come quando la candela ha un tremolio, e sta per spegnersi, o quando appaiono i fantasmi. Il testo non deve necessariamente essere stampato sul bordo inferiore del video, né deve per forza essere tutto dello stesso colore. Potreste usare anche dei bordi e degli sfondi colorati.

3. Attenti alla scrittura delle parole

Se non siete del tutto sicuri su come si scriva una parola che volete usare nel gioco, controllate su un dizionario. Il vostro computer non sa come si scrivono le parole, ed immagazzinerà in memoria qualsiasi cosa gli comunicherete. Questo potrebbe rivelarsi frustrante per il giocatore che usasse la parola scritta correttamente, ■ continuasse a ricevere messaggi d'errore perché il computer non la riconosce scritta in quel modo.

4. Distribuire i momenti d'azione

Alcuni giochi d'avventura sono un po' noiosi da giocare, in quanto tutto accade sempre nello stesso punto del gioco. Assicuratevi che vi siano cose interessanti lungo tutto l'arco del gioco.

5. Usate le istruzioni REM

Quando scrivete un programma lungo ■ complicato, qual'è un programma d'avventura, è ottima idea inserire delle istruzioni REM prima di ogni sezione. Se non lo fate, avrete buone probabilità di confondervi su quali siano le singole sezioni. Peraltro, una volta finito il programma, sarà bene che le togliate, poiché portano via spazio in memoria, rallentano il programma, ■ permettono a giocatori senza scrupoli di barare.

6. Nomi mnemonici per le variabili

Cercate di dare alle variabili dei nomi che permettono di ricordarsi facilmente che cosa sia ciascuna di esse; ad esempio OB (OBJECT) per gli oggetti, MK (MISTAKE) per gli errori, ■ così via. Se avete molto spazio in memoria, ■ con il vostro computer lo potete fare, è buona idea usare nomi lunghi per le variabili, per aiutarvi a ricordare cosa ciascuna rappresenti; esempio: al posto di V usate VERBO. Ad ogni modo, fate una lista delle vostre variabili ■ di cosa esse rappresentano, così da non confonderle mentre scrivete il programma.

7. Non complicate le cose

Non siate troppo ambiziosi con i vostri primi giochi. Un gioco semplice ■ ben pensato sarà più divertente da giocare che non uno confuso e complicato. Non tutti amano giocare con un gioco che duri diversi giorni.

8. Rendetelo comprensibile

Quando avete scritto il vostro gioco, esaminate i commenti per accertarvi che non siano ambigui o non traggano in inganno. Al posto di "TROPPO BUIO", ad esempio, potreste dire "TI SERVE UNA LUCE PER ANDARE LÀ". Ricordatevi che certe cose ovvie per voi potranno non essere affatto ovvie ad un altro giocatore. Mettete anche dei commenti spiritosi, così darete al giocatore l'impressione di un vero dialogo con il computer.

Risposte agli enigmi

Enigma poliziesco (pagina 13)

Ecco alcune delle soluzioni proposte per i problemi del gioco poliziesco. Confrontatele con le soluzioni trovate da voi.

- 1) Potrete rivelare la presenza del capello solo se darete al computer l'istruzione di esaminare il cappotto. Non potete raccogliere il capello senza avere una busta vuota nella quale metterlo.
- 2) Vi serve una chiave per aprire il cassetto, una lente per vedere il filo, ed un'altra busta vuota per mettercelo dentro.
- 3) Vi serve del gesso e un contenitore d'acqua per fare uno stampo dell'impronta sul terreno.
- 4) Vi serve del borotalco per rivelare le impronte digitali, e del nastro adesivo per asportare l'impronta così rivelata.
- 5) Vi serve un'attrezzatura portatile per l'analisi del sangue (che potete descrivere nel gioco come: scatola contenente provette e materiale da laboratorio).
- 6) Vi serve un fazzoletto per raccogliere il bastone ed una borsa di plastica per trasportarlo.

Problemi interessanti in un'avventura (pagina 15)

Ricordatevi che non vi sono risposte "esatte" a questi quesiti. Eccovi dei suggerimenti per le soluzioni.

- 1) Sollevate il tappeto e trovate una botola.
- 2) Usate il fazzoletto a protezione della bocca e del naso (nell'ipotesi che la sonnolenza sia causata da un gas nella stanza), cercate nello zaino dove troverete una bottiglia. Aprite la bottiglia e vi troverete del caffè. Bevetelo.
- 3) Leggete il papiro (che riporta un proclama per la liberazione degli schiavi).
- 4) Gettate il dolce (si tratta di un budino) in faccia al super bandito. Afferrate il telecomando e fuggite.

Quesito (pagina 28)

Ecco come potete cambiare il programma per limitare il numero di oggetti trasportabili contemporaneamente dal giocatore. Vi servono due nuove variabili, qui chiamate OT (che sta per oggetti trasportabili) e LP (che sta per: limite di possesso). Aggiungete queste alla fine della linea 70, come segue:

70...OT=0 : LP=8

Adesso dovete istruire il computer in modo che aggiunga 1 ad OT nella routine PIGLIARE, quando il giocatore raccoglie un oggetto, e sottragga 1 dalla stessa variabile, nella routine LASCIARE, quando il giocatore lascia giù un oggetto. Questo si fa aggiungendo alla fine delle linee 1010 e 1490 le istruzioni:

1010...: OT=OT+1

1490...: OT=OT-1


Ora aggiungete una nuova linea alla routine PIGLIARE per controllare se OT è pari al valore limite, prima di procedere con il seguito della routine. (LP non deve necessariamente valere 8, ma non deve valere di meno, o il giocatore non sarà in grado di portare tutti i tesori fino in fondo al gioco).

Indice analitico

- Adams, Scott, 4
ADVENTURE, 4
giochi d'avventura, diversi tipi, 4
errore di matrice, 32
matrici, 16, 17, 18, 31, 38
BASIC, 3, 4, 16, 18, 23, 36, 39, 40
BBC, 3, 23, 29, 31, 33, 36, 46
salto alle subroutine, 19, 24
uso dei caps lock, 38
matrice oggetti posseduti, 17, 18
cambiare il programma, 28, 32, 38
COLOSSAL CAVE, 4
combinazioni di parole, 22, 23
virgole per i dati, 18, 37, 45
CPU, 17
Crowther e Woods, 4
matrici bidimensionali, 17
giochi bidimensionali, 26
matrici tridimensionali, 9, 26
dati, 16, 17, 18
cicli, 18
troppo pochi, 32
immagazzinamento del, 16, 17, 18
del gioco per ZX81, 45
database, 5
debugging, 32
descrizioni, 19, 20, 25, 28, 33
descrizioni delle locazioni (matrice), 16, 17, 20, 31, 39
avventura poliziesca, 7, 11
DIM, 16, 17, 32, 44
dimensionamento, 16, 17
avventure su disco, 4
subroutine fittizia, 23
codice d'errore, 32
messaggi d'errore, predisporli nel gioco, 19, 23, 24, 25, 34
risposte, 19, 20, 25, 34
primo gioco d'avventura, 4
matrici di indicatori, 17, 18, 24, 27, 32, 44
registri indicatori, 17
Fortran, 4
oggetti trasportabili, 17
salto o subroutine, 26, 27, 34, 41
GOSUB, 19, 23, 29
GOTO, 19
grafica, 4, 46
reticolo, disegnare un, 8, 9
AIUTO, 13, 21
nascondigli, II
HOME, 33
IF ... THEN, 38
inizializzazione, 18, 19, 20, 30, 32, 33
inserimento, 19, 21, 33
analisi inserimento, 19, 22
istruzioni del giocatore, 21
variabili intere, 46
database interattivo, 5
INVENTARIO, 13
oggetti invisibili, 17, 18
LEFTS, 40
LET, 38, 40
listato del programma, 33, 37
Spectrum (Timex 2000), 38
per ZX81 (Timex 1000), 39, 45
subroutine LOAD, 30, 31
locazioni, 6, 8, 16, 17, 40, 41
numerazione delle, 8, 12
matrici delle locazioni, 16, 17, 20, 39
ciclo, 18, 20
codice macchina, 5
magia, uso della, 6
mappa del mondo d'avventura, 6, 7
schema principale, 8, 11, 12, 14, 15, 16, 26
versione per ZX81, 39
memoria, quantità utilizzata dal gioco, 28
BASIC tipo Microsoft, 3, 33
MIDS, 40
errori, correzioni degli, 32
nessuna corrispondenza, 22
matrici numeriche, 17
numerazione delle locazioni, 8
degli oggetti, 10, 12
matrice degli oggetti, 17, 18
nomi degli oggetti, 40
percorsi a senso unico, 8, 9
comandi di una parola, 21, 23, 26
ON ... GOSUB, 24, 25, 30, 32, 33, 41
errore d'intervallo, 23
troppo pochi dati, 32
condizioni d'annullamento, 19, 24, 32
penalità, 30
pianificazione, 5, 15
scopo del gioco, 6
problemi per il giocatore, 10
programma:
cambiare il, 28, 32
struttura, 19
scrivere il, 19, 24
arredi, 10, 12
QUIT, 31
READ...DATA, 18, 39
istruzioni REM, 46
RIGHTS, 40
RND, 34, 35
percorsi, 8, 9, 16, 17, 40
a senso unico, 8, 9
matrice dei percorsi, 17, 18, 20, 31
regole, 3
salvare il gioco, 30, 31
PUNTEGGIO, 3
subroutine del, 30, 36, 44
conquistare punti, 10, 30
presentazione a video, 46
computer Sinclair (Timex), 3, 18, 21, 33, 38, 40
comandi di una sola lettera, 26
abbozzo di mappa, 6
suoni, 29
Spectrum, 3, 29, 33
versione per, 38
scrivere le parole, 46
Università di Stanford, 4
immagazzinare i dati, 16, 17, 18
matrici stringa, 16
dati stringa (per Spectrum), 38
variabili stringa, 21, 22
subroutine, 19, 24, 25, 26, 31, 34, 42, 43
fittizia, 23
soggetti per giochi, 7
limite di tempo, 28
Timex 1000, 3, 18, 21, 33, 39, 45
Timex 2000, 3, 21, 33, 38
strumenti, 12
tesori, 8, 10, 11
frasi di due parole, 3, 21
oggetti utili, 10, 11, 12, 13
variabili, 16, 20, 26, 30, 33, 38, 40, 44
nomi di variabili, 46
stringa, 21
verbi, 13, 17, 24, 25, 34, 40, 43
matrice dei verbi, 17, 18
stringa per i verbi, 16, 22
VIC 20, 20, 29, 33, 34, 35, 37
muri, rivelare i, 27
armi, 12
lista di parole, 10, 12, 16
parola non trovata in memoria, 16
routine di divisione parole, 21
stringa per le parole, 16, 22
scrivere il programma, 19, 20, 21, 22, 23, 24, 25, 26, 27
casella zero (uso della), 16
ZX81 versione per, 3, 18, 21, 33, 39, 45

© Copyright per l'edizione originale Usborne Publishing Ltd — 1982
© Copyright per l'edizione Gruppo Editoriale Jackson — 1985

Impaginazione: Cristina De Venezia
Fotocomposizione: Composit - Pisa
Stampa: Grafika 78

Il nome Usborne e il marchio  sono marchi registrati dalla
Usborne Publishing Ltd., 20 Garrick Street, London WC2E 9BJ, England.

Tutti i diritti sono riservati. Nessuna parte di questo libro può essere riprodotta, memorizzata in sistemi di archivio, o trasmessa in qualsiasi forma o mezzo, elettronico, fotocopia, registrazione o altri senza la preventiva autorizzazione scritta dell'editore.

Speedy Computer

Il calcolatore è anche (o soprattutto?) una macchina divertente. Si può giocare con lui, gli si possono fare delle domande, lo si può usare per scrivere una poesia o per suonare.

Questa colorata serie di libri vi propone alcune delle cose più eccitanti che si possono fare con un calcolatore e vi spiega come farle.

Scritti in linguaggio chiaro e comprensibile a chiunque, arricchiti da una moltitudine di illustrazioni, questi libri rappresentano una spiritosa introduzione al mondo dei computer per chi comincia da zero.

costruisci programmi di ADVENTURE per il tuo computer

Ragazzi, volete partire per viaggi avventurosi, tutti quelli che la vostra immaginazione può pensare? Compratevi un computer e questo libro!

altri volumi di questa collana

APPLICAZIONI PRATICHE DEL PERSONAL COMPUTER

Un libro che vi farà scoprire piacevolmente che esiste qualcosa di diverso dai giochi, più utile e ugualmente appassionante, nel mondo dei computer.

INTRODUZIONE AL LINGUAGGIO MACCHINA

Un argomento "difficile" affrontato con linguaggio piacevole; preziose illustrazioni consentono l'immediata comprensione anche dei più complessi aspetti tecnici.

GIOCHI SPAZIALI

Chi ha detto che i computer servono solo per cose serie? Oltretutto, divertendosi si può imparare più volentieri.

COMPUTER GRAFICA

Si può dire che il computer ne sa una più del diavolo. Per scoprirlo leggete questo libro.